Community Structure Detection in Complex Biological Networks

Bennett, Laura

Awarding institution:
King's College London

Community Structure Detection in Complex Biological Networks

Laura Moore Bennett

A Thesis Submitted for the Degree of Doctor of Philosophy

Department of Informatics
Kings College London

September, 2012

Acknowledgements

I would like to thank my supervisor, Dr. Sophia Tsoka and my second supervisor, Dr. Lazaros Papageorgiou for their invaluable support and guidance throughout my Phd.

I am also grateful to the following people who have made contributions to my work: Dr. Mansoor Saqi and Dr. Artem Lysenko for providing data and comments for the work done in Chapter 6. Dr. Gang Xu and Dr. Songsong Liu for various advice regarding method implementation. Dr. Ignat Drozdov for providing the network data used in Chapter 7.

Thanks also goes to Chrysanthi who has been with me at King's since day one and has always been on hand to give advice relating to work and has become a dear friend.

Many people in the Department of Informatics at King's have offered support in various forms. There are too many people to mention by name, but I am grateful to them all. I would like the thank in particular Golnaz and Carl for kindly reading chapters of my thesis. Thanks also goes to Gbolahan for being my "study buddy" during the final weeks.

I would also like to thank the following two people: Erica, for her support in so many aspects of my life during the last four years and Anne, who has offered invaluable advice and encouragement throughout my phd.

I also want to thank my Mum, my brother, David, and my sister-in-law, Rowie, for their endless encouragement and support.

Finally, this thesis is dedicated to the memory of my dad, Charlie Bennett.

Abstract

With the advent of high-throughput technology there has been a large increase in the availability of biological data, such as interaction data of genes, proteins and metabolites. It is therefore necessary to develop ways in which these data can be efficiently modelled and analysed. Networks offer a natural modelling framework for complex biological systems and as such, network theory and related computational approaches have proven important in bioinformatics. A particular facet of network theory that has been employed to analyse biological networks is community structure detection. Community structure is a modular network topology where communities are defined as groups of nodes with dense intra-community connections and less dense inter-community connections. Methods to uncover such communities in complex biological networks have the potential to contribute towards a better understanding of the underlying organisation of a system. Consequently, this thesis focuses on the development of a series of mathematical programming models to address various manifestations of the community structure detection problem. The aim is to produce more information-rich models that can accurately represent the features of biological systems; with weighted and unweighted interactions, disjoint and overlapping communities and network dynamics all being considered.

First, the detection of disjoint communities in unweighted networks is approached through a two-stage procedure, known as iMod. A mixed integer nonlinear programming (MINLP) model optimises modularity to find an initial partition which is then improved by iteratively solving a mixed integer quadratic programming (MIQP) model. A comparative analysis shows that iMod finds globally optimal solutions for networks of up to 512 nodes and outperforms all other methods tested when applied to larger networks. Subsequently, the MINLP model is generalised to weighted networks, known accordingly as WeiMod. Competitive results are found when WeiMod is compared with several other well known methods from the literature. Next, the work on disjoint community structure is extended to find overlapping modules. An MINLP model, known as OverMod, transforms disjoint to overlapping communities through the optimisation of the community strength metric. OverMod is compared with similar methods from the literature and is further assessed on protein-protein interaction (PPI) networks to test the method's ability to extract meaningful biological results. It is shown that proteins assigned to more than one module exhibit topological and functional properties indicative of their strategically important role in the organisation of the PPI networks. The work on disjoint and

overlapping community structure concludes with the investigation of a network generated from sequence, protein interaction and co-expression data, for the fungal pathogen, *Fusarium graminearum*. The functional coherence of communities, properties of multi-clustered genes and aspects of virulence-associated genes are all explored in an attempt to link topological and functional features.

Finally, the concept of community structure in dynamic networks is explored. Consensus clustering is tackled; defined as detecting a single partition of a dynamic network that is relevant across multiple snapshots. This is addressed by extending the previously proposed MIQP and MINLP models such that average modularity across network snapshots is now optimised. A comparison is made with a similar method from the literature showing that the proposed approaches achieve competitive results for small to medium sized networks.

Overall, this thesis demonstrates that the flexible nature of mathematical programming lends itself well to developing versatile solution procedures for community structure detection. The method evaluations show the proposed algorithms to be comparable to other approaches from the literature and able to detect meaningful results in biological applications. Finally, the methods described in this thesis have the potential to infer important topological-functional relationships and help to provide insight into the organisation and evolution of biological systems.

Publications

During the compilation of this thesis, the following related articles have been published by the author.

- Gang Xu, **Laura Bennett**, Lazaros G. Papageorgiou and Sophia Tsoka (2010) Module detection in complex networks using integer optimisation. Algorithms for Molecular Biology, 5, 9880-9885.
- **Laura Bennett**, Songsong Liu, Lazaros G. Papageorgiou and Sophia Tsoka (2012) Detection of Disjoint and Overlapping Modules In Weighted Complex Networks. Advances in Complex Systems, 15, 1150023.
- **Laura Bennett**, Songsong Liu, Lazaros G. Papageorgiou and Sophia Tsoka (2012) A mathematical programming approach to community structure detection in complex networks. In: I.D.L. Bogle and M. Fairweather, M. (Eds.) 22nd European Symposium on Computer Aided Process Engineering (ESCAPE-22), Computer Aided Chemical Engineering, vol. 30. Amsterdam, Elsevier. pp. 1387-1391.
- **Laura Bennett**, Artem Lysenko, Lazaros G. Papageorgiou, Martin Urban, Kim Hammond-Kosack, Chris Rawlings, Mansoor Saqi and Sophia Tsoka (2012) Detection of multi-clustered genes and community structure for the plant pathogenic fungus Fusarium graminearum. In: D. Gilbert and M. Heiner (Eds.) CMSB 2012, LNCS 7605, pp. 69-86.
- Artem Lysenko, Martin Urban, **Laura Bennett**, Sophia Tsoka, Elzbieta Janowska-Sejda, Chris Rawlings, Kim Hammond-Kosack and Mansoor Saqi (2012) Network-based data integration for predicting virulence proteins in the cereal infecting fungus Fusarium graminearum (submitted).

Contents

Chapter 1

Introduction

1.1 Overview

With the advent of high-throughput technology, such as microarray technology and yeast two-hybrid screening, there has been a large increase in the availability of biological data, such as interaction data of genes, proteins and metabolites. As a result, it is necessary to develop new ways in which these data, which can exist on a very large scale, can be efficiently modelled and analysed.

Networks offer a natural modelling framework for complex biological systems. Many biological systems can be represented as a series of interactions, where nodes represent biomolecules, such as genes or proteins, and interactions may represent physical or functional associations. Such interactions can be inferred from gene expression experiments, sequence similarity or even from searching biomedical literature. Once a network representation of the system has been generated, analytical tools are required that can process the data and extract meaningful biological information.

In bioinformatics, network theory and related computational approaches have proven important in the investigation of biological systems. A particular facet of network theory that has been employed to analyse biological systems is community structure detection. Community structure is a modular architecture common to complex systems where communities or modules are defined as groups of nodes with dense intra-module connections and less dense inter-module connections. The members of such communities usually share common characteristics or properties or work together towards a particular goal.

Biological networks have been shown to exhibit community structure [28, 74]. A modular topology confers benefits such as multi-functionality, robustness and 'evolvability' of biological systems [128, 227]. The evolution of community structure in biology has resulted in the semi-independence of functional units of biomolecules, allowing changes within modules to occur with minimal disruption to the function of the whole system, thus promoting robustness [90]. It has also been found that functional modules are conserved across species, illustrating the role of this topology in the functional evolution of biological systems [63].

Consequently, identifying communities in biological systems has contributed towards a better understanding of a biological system as a whole. For example, the communities identified in a protein-interaction network may represent groups of proteins with similar function [43, 189] or the gene products of genes responsible for similar diseases [75, 157]. Identifying such communities could lead to the deduction of missing information, such as protein function or to the identification of potential drug targets. In general, interpreting biological systems as networks and decomposing these networks into modular sub-networks provides a more global view of a biological system and not just its individual components. Investigating the mechanisms of biological systems in such a way can help bridge the gap between molecular and modular biology. Consequently, the development of methods that efficiently and accurately identify community structure play an important role in bioinformatics applications.

The detection of modules or communities has become widely accepted as a means of revealing the underlying properties of complex networks and the development of associated methodologies has featured in many different research areas. A major breakthrough in community structure detection methodology was the introduction of a measure known as modularity; this measure expresses how well defined the community structure of a particular partition of a network is [148]. Since then, modularity optimisation via a variety of optimisation techniques has been the basis of many clustering algorithms. The global optimum value of modularity represents the configuration of a network into communities with the most dense intra-module connections and the least dense inter-module connections. However, modularity optimisation is known to be NP-hard [35] and therefore most methods are devised with the aim of finding good near-optimal solutions with reasonable computational cost.

The goal in community structure detection method development is not only to create

accurate solutions, but also to improve network models through the inclusion of more informative features of complex systems. The standard form of the problem is to partition an unweighted, undirected network into a series of disjoint communities. However, this definition may not sufficiently capture the system under study or meet the application requirements and therefore additional information or constraints may lead to more accurate solutions. Such additional features or criteria may come in the form of weighted or directed interactions, allowing nodes to belong to more than one module or incorporating network dynamics. Consequently, community structure detection has a continually evolving problem statement with the aim of producing more realistic representations of complex systems. Mathematical programming is a modelling framework suited to such method evolution with the ability to meet the changing needs of the various manifestations of the community detection problem. The flexible nature of mathematical programming lends itself well to developing versatile solution procedures.

This thesis focuses on the development and application of mathematical programming based community structure detection algorithms. Different forms of the problem are tackled; detecting disjoint and overlapping modules in weighted and unweighted networks and characterising the community structure of dynamic networks. Each stage of the method development is addressed through the evolution and adaptation of optimisation models. The evaluation procedures undertaken show the methods to be comparable to other approaches from the literature and able to detect meaningful results in biological applications. The series of optimisation models described in this thesis have the potential to infer important topological-functional relationships and help to provide insight into the organisation and evolution of biological systems.

1.2 Research aims

The overall aim of this thesis is to develop a series of mathematical programming methods to tackle the community structure detection problem with a view to aiding the extraction of biologically meaningful results from biomedical data. This aim can be further decomposed into five core research goals:

- To build on existing modularity optimisation methodology to detect disjoint communities in larger scale as well as weighted networks.

- To define and implement a solution procedure to the problem of detecting overlapping community structure.
- To define and implement a solution procedure to the dynamic community structure detection problem.
- To evaluate methodology to show comparability with existing methods from the literature.
- To demonstrate the potential of such methods to find meaningful results in biological applications.

1.3 Thesis outline

The thesis will unfold as follows. Chapter 2 provides background information on the topic of complex networks, community structure detection and its significance in biological systems. Related work is discussed followed by a review of existing modularity optimisation methods. In particular, a description of a mixed integer quadratic programming (MIQP) model is given, which due to achieving global optimality has scalability limitations. This method represents the starting point for the methodologies derived in this thesis. In Chapter 3, the MIQP method is incorporated into a two-stage procedure for detecting disjoint communities with the aim of clustering larger scale networks than previously. Stage 1 formulates modularity optimisation as a mixed integer nonlinear programming (MINLP) model and Stage 2 involves the iterative application of the MIQP model. A comparative study is carried out to evaluate the method's performance across several other modularity optimisation methods on a series of benchmark networks. Chapter 4 goes on to generalise the MINLP model to detect disjoint community structure in weighted networks and again a comparative study with existing methodology from the literature is carried out.

Chapter 5 extends previous methodologies to tackle the detection of overlapping community structure in a two-stage procedure. First, a partition of the network into disjoint communities is detected. The disjoint communities are then transformed to overlapping communities via a further MINLP model that optimises a metric known as community strength. This method is evaluated in an application to protein interaction networks of two well-annotated organisms, rat and human, where the properties of proteins belonging to more than one module are investigated. The method's potential is further

investigated in Chapter 6 in an exploratory analysis of an integrated network of a fungal pathogen with a largely unannotated genome. In particular, focus turns to the characterisation of virulence associated genes in the context of community structure.

In Chapter 7, a further development is considered through the incorporation of network dynamics into the community structure detection problem. The previous MIQP and MINLP models described in Chapters 3 and 4 are extended to cluster time series network snapshots simultaneously and are evaluated through a comparative analysis with a similar method from the literature.

Finally, Chapter 8 concludes the thesis by first giving a brief overview of each chapter. Next, the research aims outlined above are revisted in order to indicate where they are addressed in the thesis and to assertain to what degree they are fulfilled. Major contributions of the thesis are then given, followed by discussions regarding the limitations of the work and the potential avenues of future research.

Chapter 2

Background and related work

The previous chapter outlined the rationale behind the work that will be presented in this thesis and set out several key goals. The central aim of this work is to develop a series of mathematical programming methods to tackle the community structure detection problem with a view to aiding the extraction of biologically meaningful results from biomedical data. In order to carry out this task, a thorough understanding of the nature of the problem, the potential applications and the existing methodologies as well as the possible limitations is required. In this chapter, a review is given of essential background and related work that underpins this thesis. This includes introducing the topic of complex networks and their properties and describing the biological systems that these networks can model. From the properties of complex networks, community structure is then taken forward and discussed in more detail followed by illustrative examples of associated key applications in bioinformatics. Finally modularity, a measure of the degree of community structure exhibited by a complex network, is discussed and an overview of clustering algorithms based on modularity optimisation is given. Overall, the information provided in this chapter sets the scene for the starting point of the research described in the remainder of this thesis.

2.1 Introduction to complex networks

Complex networks can arise from many real world situations such as social interactions, the Internet, scientific collaborations and biological systems. Despite their obvious differences in terms of constituent nodes and interactions, these networks have many similarities in terms of organisation and other topological properties. In order to understand and analyse these individual networks, it is necessary to understand the nature of complex networks in general as much of their underlying properties are governed by generic organisation principles. In this section, properties common to most complex networks are described. Furthermore, examples of biological systems that can be modelled by complex networks are given.

2.1.1 Properties of complex networks

A network is a graph made up of nodes, which represent some type of entity, joined together by edges, which represent the associations between the different entities. Throughout this thesis, the term vertices will be used interchangeably with nodes, and links or interactions with edges. Complex networks exhibit certain properties including scale-free distribution of nodes, high clustering coefficients, the small-world property and community structure [74]. These characteristics and other essential concepts associated with complex networks are described below.

First, the most simple of network measures is node degree, k, the number of edges a node makes with other nodes in the network. This local measure can be transformed to a global measure of a network by calculating the average node degree of all nodes in the network. The probability of finding a node in a complex network with a higher than average degree is greater than in a random network where the probability of having an edge between a pair of vertices is equal for all possible pairs. Node degree distribution, $P(k)$, is also commonly used in network description, where $P(k)$ is the number of nodes in the network with degree k.

A major step towards the understanding of the nature of complex networks was the discovery that their degree distribution follows a power-law [28]:

$$P(k) \sim k^{-\gamma} \tag{2.1}$$

Where generally, $2 < \gamma < 3$. Consequently the networks, known as scale-free networks, have a relatively small number of highly connected nodes (hubs), with the remaining nodes having relatively few connections, illustrated in Figure 2.1. A mechanism known as preferential attachment is responsible for the emergence of this scale-free topology [28]. This is where networks grow through the connecting of new nodes to existing nodes with a higher probability of linking to a node with a large degree, analogous to the "rich get richer" scenario. As a result of the scale-free degree distribution, complex networks are robust to random perturbations, but weak to targeted attacks on hub nodes [17]. This robustness is particularly important to biological systems since it promotes resilience against random mutations [53].

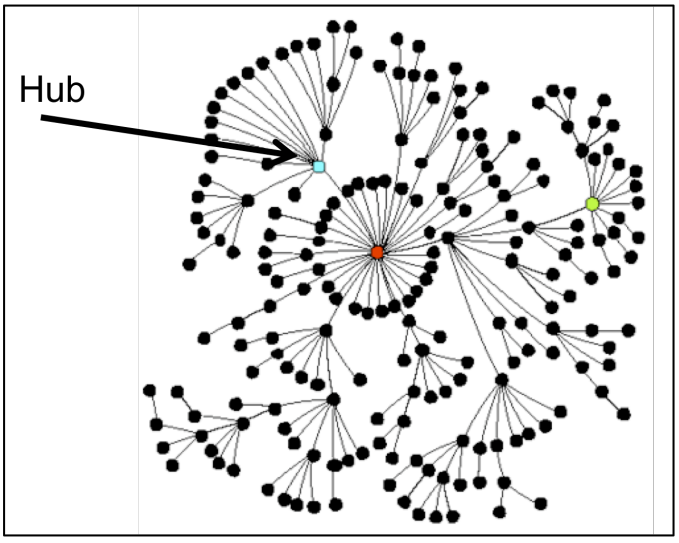

FIGURE 2.1: Visualisation of a scale-free network where hubs and the effect of preferential attachment can clearly be seen. Figure taken from [122].

Other measures used to describe complex networks include shortest paths between two nodes and the average shortest path length for an entire network, giving an indication of the number of steps required to cross a network. If a network can be traversed in a small number of steps, it is said to exhibit the small world property, commonly known as the "six degrees of separation" [138]. More specifically, in a small world network, the distance L between two randomly chosen nodes grows proportionally to the logarithm of the number of nodes N in the network [210]:

$$L \propto \log N \tag{2.2}$$

The average shortest path length for complex networks is smaller than in random networks.

Furthermore, in complex networks, two vertices that are both connected to a third common vertex have a higher probability of also being connected to each other than in a random network [74]. This degree of connectivity is measured by the clustering coefficient of a network, a measure that is defined for both individual nodes and for the entire network. The clustering coefficient of a node reflects the degree of connectivity between its immediately connected neighbours and is defined as the number of links between the nodes within the neighbourhood divided by the number of possible links between them. For each node, i, in a network, the clustering coefficient is defined as follows [210]:

$$C_i = \frac{2N_i}{d_i(d_i - 1)} \tag{2.3}$$

Where N_i is the number of links between the immediate neighbours of node i and d_i is the degree of node i. C_i indicates how close the neighbourhood of node i is to being a clique (i.e. a fully connected graph). Again, this local measure transforms into a global measure with the average clustering coefficient quantifying the degree of clustering for the whole network [210].

Finally, complex networks exhibit a property known as community structure, where there are high concentrations of edges within groups of vertices, and low concentrations between these groups [148]. Community structure is the main topic of this thesis. An example of community structure is given in Figure 2.2 where three groups of densly connected nodes are shown to be loosely connected by a few edges. These groups, known as modules or clusters or communities, terms which will be used interchangeably throughout this thesis, comprise nodes that generally share common properties or characteristics. Many such groupings exist in society, for example schools, workplaces, families, villages, countries etc. Furthermore, communities exist as web pages with similar topics in the network of the World Wide Web [68], scientists with similar research interests in collaboration networks [144] and proteins with similar function in protein interaction networks [189]. Studying community structure allows for the classification of nodes and can also uncover underlying properties about the organisation and functionality of a system. Consequently many algorithms for the detection of communities in complex networks have been developed in recent years. A deeper discussion about aspects of community structure detection, its applications and some of the existing algorithms will feature in Sections 2.2 and 2.3. First however, the other main theme of

this thesis is the application of community structure detection methods in biological networks, therefore next a brief introduction is given to the types of biological systems that exhibit the properties described above and are thus commonly modelled by complex networks.

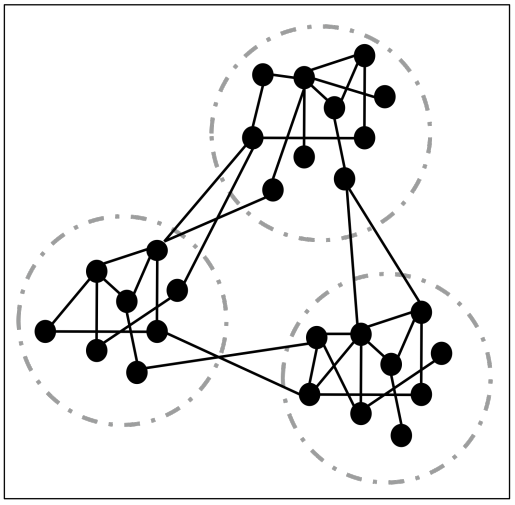

FIGURE 2.2: Example of the modular topology of community structure. Dashed lines represent the border of each module.

2.1.2 Modelling biological systems with complex networks

Biological systems take many different forms, from food webs in ecology to the representation of groups of various biochemical reactions in the cell. In recent years, with the advent of high-throughput technology, such as microarray technology and yeast two-hybrid screening, there has been a large increase in the availability of biological interaction data. Networks can be built from measures of sequence similarity, genetic interactions, gene co-expression, protein interactions, domain interactions and term co-occurrence in the scientific literature. These complex data can exist on an extremely large scale and therefore requires efficient tools to facilitate its analysis. Complex networks offer a natural modelling framework for the conceptualisation of the intricate relationships that exist in biological systems. It has been shown that many biological systems exhibit the properties of complex networks described in the previous section [74]. As such, network theory and related computational approaches have proven important in the investigation of biological systems. Below a brief description is given of

three of the main types of biological systems that can be effectively modelled by network representations:

- *Gene regulatory networks.* Transcription factors control the activation or inhibition of the transcription of genes to mRNA. Transcription factors are themselves products of genes and therefore in essence, genes are regulating each others' expression. These pairings of genes can be modelled in gene regulatory networks with nodes representing genes and directed edges indicating the direction of flow from transcription factor to gene, as shown in Figure 2.3. Transcription regulatory networks have been shown to exhibit a scale-free topology [18].

- *Protein interaction networks.* Protein-protein interactions (PPIs) occur when two or more proteins bind to carry out their biological function. There are a multitude of methods available to detect these interactions, including Co-immunoprecipitation, yeast two-hybrid screening and tandem affinity purification (TAP). PPI data can be retrieved from databases such as BioGrid [191], STRING [194], BIND [27] and DIP [217]. In PPI networks nodes represent proteins and edges represent physical interactions, i.e. binding. Unlike gene regulatory networks, these mutual interactions can be modelled with undirected edges. PPI networks have also been shown to be scale-free networks [141].

- *Metabolic networks.* A large number of metabolic reactions occur at any time in living cells and the product of one reaction is usually used by another, thus metabolic reactions are strongly interconnected and form metabolic pathways and networks. A metabolic reaction is the transformation of chemical substances or metabolites (reactants) into other substances (products) usually catalysed by enzymes. The metabolic network of a particular cell or organism is the complete network of metabolic reactions. Metabolic reactions have been modelled at various levels of complexity. For example, a metabolic network can involve three types of nodes to represent metabolites, reactions and enzymes with two types of directed edges, one to represent mass flow and the other for catalytic regulation [16]. However, more simplified versions of the metabolic map also exist. For example, the substrate network, where nodes representing reactants are linked by an undirected edge if they occur in the same reaction [202]. Alternatively, networks where nodes are reactions and edges join reactions if they share at least one metabolite [16]. All of the above network representations have been shown

to be scale-free [94, 195, 202]. Metabolic pathways can be retrieved from several databases including EcoCyc [97] and MetaCyc [39].

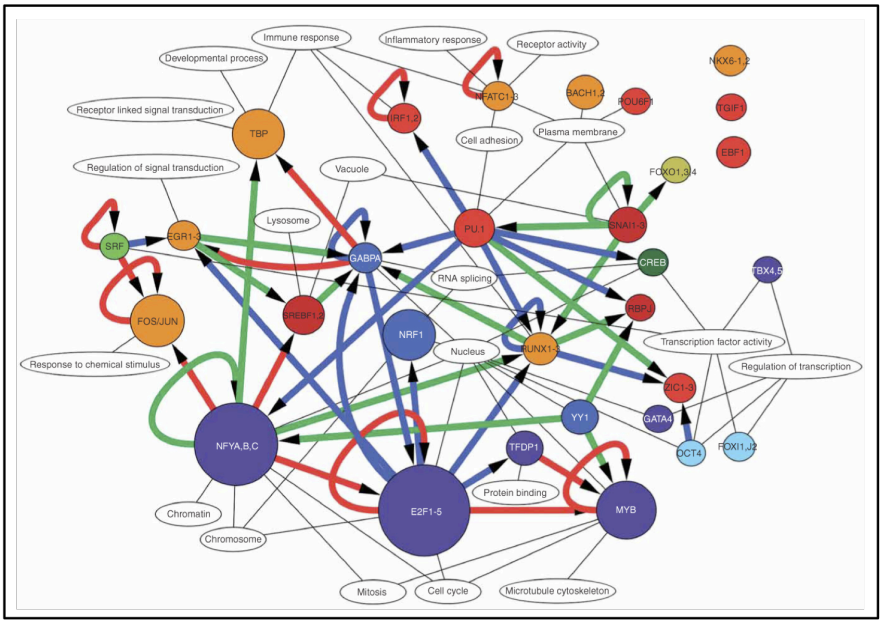

FIGURE 2.3: Gene regulatory network example: the predicted core transcriptional regulatory network of the human monocytic cell line THP-1. The arrowed edges indicate the direction of transcription factor to gene. Figure taken from [40].

In addition to the above networks, biological networks also include (i) domain networks with domains as nodes and co-occurrence in proteins as links, (ii) phylogenetic trees, with taxons (species, genes or DNA) joined based on DNA or protein similarity and (iii) gene expression networks with edges based on the correlations between gene expression profiles. Furthermore, with the many types of biological data being modelled by a single framework (i.e. complex networks), it is possible to produce integrated networks with nodes and interactions from multiple data sources, thus obtaining a wider view of the structure of the whole cell [132].

In many cases the networks described above comprise thousands of nodes and interactions and therefore their visualisations do not prove informative. For example see the *Caenorhabditis elegans* PPI network, an example of a hair ball network, in Figure 2.4. Consequently analytical tools are required to break down the network into more manageable sized units to better comprehend the underlying topology and organisation which can in turn can lead to a better understanding of how cellular processes are coordinated.

A higher level interpretation of a biological system can be provided through analysing its community structure and it follows that the development of accurate algorithms for uncovering modular topology is an important task. Community structure detection and its application to biological networks are explored further in the next section.

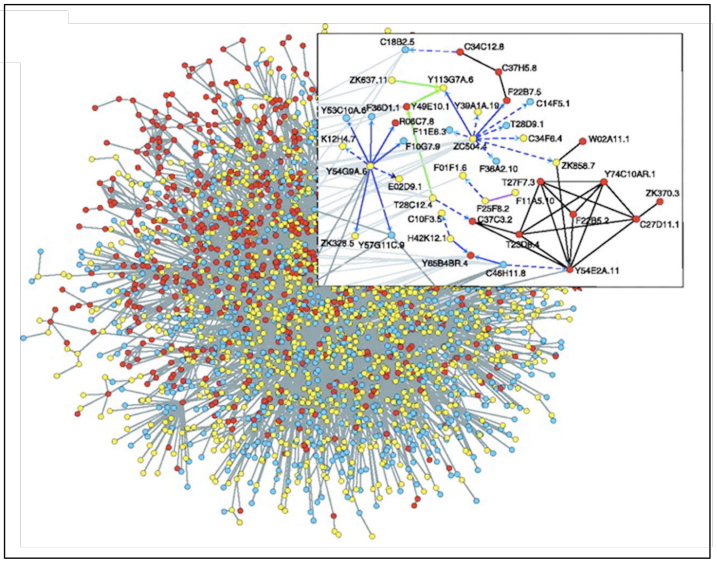

FIGURE 2.4: Example of a PPI hair ball network: *C. elegans* protein interaction network where the colour of the nodes indicates the protein's phylogenetic class. Figure taken from [16].

2.2 Community structure in complex networks

Community structure was previously introduced in Section 2.1.1 as a property of complex networks. Here the concept is discussed in more detail in terms of its general applicability and its significance in biological systems, followed by several key examples of its application in bioinformatics.

2.2.1 The community structure detection problem

Community structure detection has appeared across several disciplines including, social science, computer science and biology, as a means of analysing complex data and extracting information about the underlying organisation of a system. The most basic version

of the problem statement is to find the partition of a network into disjoint communities with the most dense intra-module links and the least dense inter-module links. Many methods to detect community structure exist, with the aim of finding dense groups of nodes based purely on network topology. The assumption is that the entities assigned to the same module based on topological features will also share some non-topological features. Information taken from the modules derived and the commonality between members can be used in classification, prediction or assignment of some form of role in the overall system. Some brief examples are given below.

First, in [167] a network of on-line shoppers, with connections representing co-purchasing of items, was partitioned into communities representing groups of people with common interests in books, music or films. This information was then used further to recommend purchases to buyers based on those made by people with similar tastes. Furthermore, clustering techniques have identified functionally coherent modules in the yeast PPI network, grouping proteins with the same or similar biological functions together [43]. The modules were then used to make predictions of unknown functions of some genes based on the module they belonged to and the functions of its constituent members. A final example can be seen in the clustering of a network modelling the spread of disease based on host interactions [179]. Individuals that were found to bridge more than one community were identified as potential immunisation targets due to their strategic role in the network. Furthermore, they were found to be more effective targets than simply targeting highly connected individuals.

These simple examples give an idea of the potential applications of community structure detection in various areas of research, where, despite obvious differences in subject matter, the fundamental goal is the same. Next, a well-known motivating example is described to better illustrate the nature of the problem.

2.2.2 Motivating example: the Zachary Karate Club network

The Zachary Karate Club network [224] is a well-studied network often used in benchmarking tests for community structure detection method development. The network models the relationships between 34 members of a karate club in an American university in the 1970s, where ties were determined based on the number of situations in and outside the club in which interactions occurred over a period of three years. Due to a dispute between the club president and the instructor over the price of karate lessons,

the original karate club dissolved and two separate clubs were formed according to the loyalties of the members and the existing friendships.

The aim of any method attempting to identify the community structure of the network would therefore be to find two modules representing the two newly formed clubs. Since such methods cluster networks based on topology alone, it is desirable that the known partition corresponds to that with the most dense connections within communities and the least dense connections between communities. Therefore a method's performance can be evaluated according to how close the predicted community structure is to the network's true community structure.

Figure 2.5 (a) shows the full network, with Figure 2.5 (b) showing the known community structure by assigning a colour to each node according to its community membership. Nodes 1 and 34, representing the instructor and the president respectively, are highlighted. These nodes are also the two most connected nodes in the network, reinforcing their crucial role in the split. This small example serves to illustrate the basic idea behind the community structure detection problem, while simultaneously allowing the concept to be envisaged on a larger scale, e.g. detecting known protein complexes in protein interaction networks. It follows that focus next turns to the significance of community structure in biology and specific applications where important insights into biological systems have been derived.

2.2.3 Community structure in biological systems

Modular structures exist at all levels of biology from the molecular structure of individual genes to the body plans of whole organisms to entire ecosystems. This section focuses on the significance of community structure in biological networks at the cellular level; why do they exhibit this topology and what benefits does this architecture confer to the overall functioning of the system?

First, community structure can be thought of as a general design concept in engineering. For example, building complex systems out of simpler components, i.e. modules, is desirable as the individual components can be tested independently before being integrated as a complete system. Furthermore, the independent modules carrying out a specific function as part of a larger global function can be used in more than one system, therefore reducing developmental time. Finally, if one part breaks down, only this part needs to be repaired, making the maintenance of such systems easier. Each of

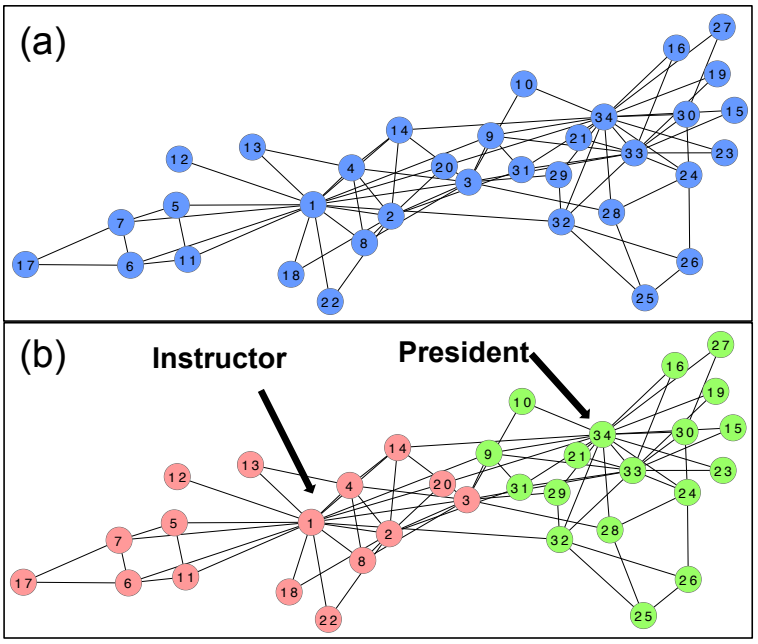

FIGURE 2.5: Karate club network example: (a) the full network; (b) the community structure of the network, where each colour represents one of the two clubs that were formed following the breakdown of the original karate club. The instructor and the president, nodes 1 ad 34 respectively, are identified as the ringleaders of the split.

these factors contribute towards efficiency in design. A modular topology is an efficient topology. However, biological systems are not designed, they evolve and therefore the question can be asked if this design concept has been chosen by evolution for the same reasons; does this efficient design also apply in biology?

Lorenz et al. [128] hypothesise that selective pressures lead the changing environment in the cell to adopt adaptable frameworks, and in competition among different evolutionary frameworks, the most efficient dynamics are provided by a modular topology. A network with community structure comprises components that are internally coherent in terms of density of links and ideally in terms of function and that are relatively autonomous to the rest of the network. This leads to systems that are simultaneously both robust and flexible, characteristics beneficial to the functioning of biological systems. For example, the robustness of the system is demonstrated in the case where a defect occurs within a module, e.g. a random mutation. Generally, the effects of the change remain local and consequently the overall functioning of the entire system is minimally changed [128]. At the same time, the relative independence of the modules is conducive to the evolution of a biological system as individual components can evolve independently and as long as

the communication between modules remains consistent, overall functioning can continue without disruption [128]. However, it is also noted that the scale-free architecture results in biological systems being weak to targeted attacks on hubs [53].

This is not the only theory offered as an explanation of the emergence of modularity in biological systems. For example, it has been suggested that it may be as an effect of gene duplication [87] or that it is a result of horizontal gene transfer [165, 47] or indeed that it has in fact no biological significance [208]. Despite difference in opinion regarding the reason for its existence, it is generally agreed that community structure is present in biological networks and that it can act as a powerful aid in research. A few examples that illustrate the potential of community structure detection to uncover meaningful results in biological networks are discussed next.

2.2.4 Applications of community structure detection in bioinformatics

In this section three examples of the application of community structure detection to biological networks are described in order to illustrate the relevance of this analytical approach in bioinformatics. Focus is not yet on the methods used to detect the modules, but rather on the variety of applications and the ability of the approach to gain insights into biological systems.

2.2.4.1 Evaluating the functional coherence of modules

Here, a study by Chen and Yuan [43] on the yeast PPI network is described as it features many common aspects associated with community structure detection applications in biological networks. These include network generation through the integration of multiple datasets, using experimental information to derive interaction weights and the role of community structure detection in function prediction. Most importantly however, this study illustrates one of the most common goals, if not the overall goal, of community structure detection applications: the detection of biologically coherent modules. Furthermore, the analysis features means of both topological and functional evaluation of module coherence. A flow chart outlining the various stages of the study is shown in Figure 2.6.

First, various yeast PPI datasets generated from high throughput techniques were downloaded, with each interaction assigned a confidence score. The yeast PPI network was

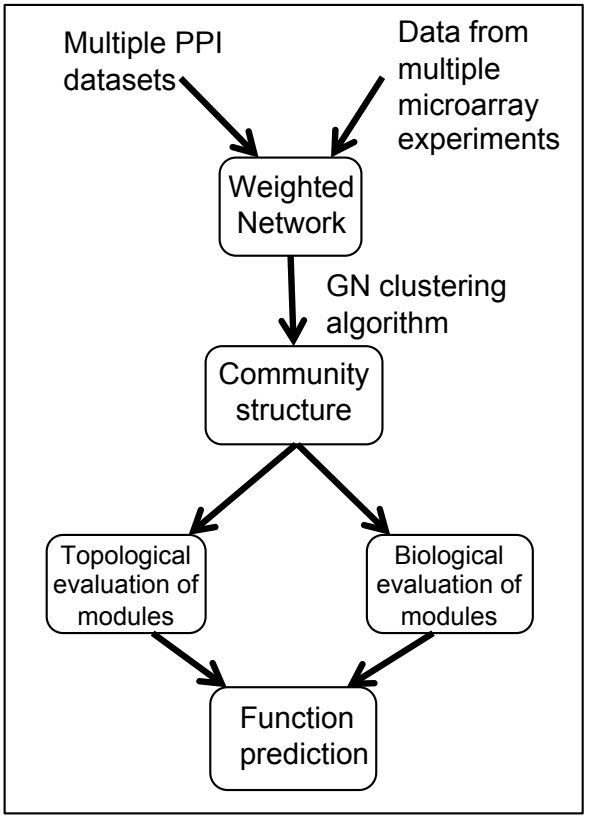

FIGURE 2.6: Outline of the network analysis study by Chen and Yuan [43]: network construction, community structue detection, module evaluation and function prediction.

constructed by combining all high confidence interactions from all datasets, resulting in a binary network of 3409 nodes and 10899 edges.

One of the main aims of the application of network analysis tools in biology is to create more realistic and informative representations of the systems under study. A first step towards this is including weighted interactions to quantify the strength of association between nodes, rather than a simple binary representation. In this study, weights were generated with information gathered from 265 microarray experiments, downloaded from the *Saccaromyces* Genome Database (SGD) [45]. Raw expression change ratios, r, were transformed to z-scores in order to make comparisons across experimental datasets. It followed that the z-score of a given gene g in microarray experiment m that is changed by the ratio r was defined by:

$$Z_g^m = \frac{(r - \mu)}{\sigma} \tag{2.4}$$

where μ and σ are the mean and standard deviations of the change ratios in experiment m. The edge weight between two proteins i and j was then defined as the average of the z-score differences over all experiments:

$$w_{ij} = \left| \frac{1}{n} \sum_{m=1}^{n} (Z_i^m - Z_j^m) \right| \tag{2.5}$$

where n is the number of microarray datasets used, 265 in this case. w_{ij} therefore represents the dissimilarity between the expression profiles of the two genes, corresponding to the distance between two nodes in graph theory.

The weights were mapped to the PPI network which was then partitioned using the pioneering GN algorithm [148]. The method is based on "edge betweenness", a measure calculated by the total number of shortest paths that traverse the edge in question (details of the full algorithm are given in 2.3.1). Chen and Yuan extended the betweenness measure so that the shortest path calculations were based on the newly generated edge weights. The GN algorithm detected a partition of the yeast PPI network into 266 modules ranging in size from 5 to 98 nodes. Each of the observed modules were then validated topologically and functionally as described below.

A measure, known as connectivity density was proposed in order to validate the modules from a topological perspective. If the in-degree of a node is defined as the number of connections it makes with nodes in its own module, then connectivity density of a module is defined as the ratio of total in-degrees to the total number of connections, where the lower the density, the less likely the observed module is a true module in the topological sense. It was found that the observed modules were more densely connected than randomly generated control modules, thus validating the authenticity of the modules and confirming their suitability as candidate functional modules.

The observed modules were then validated in terms of their biological significance. First the phenotypic similarities between the nodes in a module were used as a measure of functional coherence. It is assumed that a module performs a relatively coherent biological function, and therefore it is expected that knocking out any of the genes in a module will produce a similar phenotype. Here, each gene was represented by a phenotype vector of length 31 (corresponding to 31 experimental conditions) and the Euclidean distance was used to calculate a measure of the phenotype difference between two genes. The average difference of all pairs of genes in a module was used to calculate the total phenotype divergence of a module, therefore assigning to it a

measure of functional coherence. It was found that 185/254 modules had phenotype divergence lower than the average phenotype difference of all the yeast open reading frames (ORFs). Again, the significance of the results were confirmed by comparing with 20 randomisation experiments, where overall the controls had a higher phenotype divergence than the observed modules.

The modules were then compared with known protein complexes from the Comprehensive Yeast Genome Database (CYGD) at MIPS [86] to further evaluate their biological coherence. Each protein complex was matched against the observed modules and the maximum overlap was calculated. Results showed that known protein complexes were to a large extent contained in the observed modules, with many of them identified in their entirety. Again the results were shown to be significant through randomisation experiments.

These are two examples of the multiple ways of assessing the level of functional coherence of a module. A few others are described below.

First, validation can be based on functional homogeneity of a module. The aim is to determine whether any functional categories (e.g. KEGG pathways [96], GO terms [23]) are over-represented in the set of genes or proteins in a module. The hypergeometric distribution has been employed to find the probability that a given set of proteins/genes is enriched by a function purely by chance, through comparison with a reference set of proteins/genes (e.g. the whole genome of the organism in question). The hypergeometric distribution gives a p-value for each individual module in a partition for each functional category tested, and since in general more than one category is tested, these p-values need to be corrected for multiple testing (e.g. using the Benjamini-Hochberg procedure). Measures, such as the Clustering Score [204], have been proposed to determine the functional homogeneity of an overall partition and not just the individual modules. Consequently, such measures can be used to compare partitions of the same network found by different methods.

Alternatively, in [132] the Average Information Content of the Most Informative Common Ancestor (AIC-MICA) is proposed to gauge the degree of commonality of the proteins in a particular module based on Gene Ontology (GO) annotations [23]. The GO project offers descriptions of gene products in three structured controlled vocabularies (ontologies): Biological Process (BP), Molecular Function (MF) and Cellular Component (CC). Each ontology is structured as a directed acyclic graph (DAG), with general terms at the root of the graph, with annotations becoming more specific as one

moves lower down the graph. For example, a broad molecular function term is "catalytic activity", whereas a more specific term on the same branch as catalytic activity is "adenylate cyclase activity". Each protein can be annotated with multiple GO terms for each of the three categories. The information content (IC) of an annotation depends on its position in the DAG: the closer to the leaves the higher the IC.

The AIC-MICA method identifies a set of representative Most Informative Common Ancestor (MICA) terms for a module. For each GO term in the ontology, the number of proteins in the module assigned the annotation is counted. A coverage threshold is chosen, where only GO terms associated with at least a specified percentage of proteins in the module are retained, with any redundant terms removed. A trade off is then made between the level of coverage of a term and the specificity (degree of IC) of that term. For example a certain GO annotation may be applicable to 90% of the proteins in a module, but it may be a very general term with low IC, e.g. "cellular process". There may be however, a GO term much lower down the tree, with a higher IC that only covers 70% of the proteins in the module. The MICA term is defined as a GO term, annotated to a set of proteins (a module), which has the highest possible IC value. The AIC-MICA statistic itself, given to each module, is an average of their MICA values, which serves as an indicator of annotation coherence at different levels of coverage. A higher value would indicate that most of the MICAs for the set are found lower in the ontology and therefore commonality in annotation is at a level with higher specificity. The average of AIC-MICA statistic for all clusters in a partition can be taken as a value of functional coherence for the whole partition and is comparable across partitions of the same network. The AIC-MICA method is employed to validate a network partition in Chapter 6, Section 6.3.1.

The functional validation methods described above are two ways in which one can evaluate the modules in a partition of a network. The aim is not only to ensure that the modules found are biologically valid, but also to assign one or more functions to a module to indicate the biological purpose of the group of proteins or genes. This assignment of a consensus function or functions can then be used to predict potential functions for genes that are as yet unannotated.

In the Chen and Yuan study [43], one module in particular was chosen from the partition of the yeast PPI network to illustrate how function prediction can be carried out. In a module of 18 proteins, 11 were annotated by CYGD as playing a role in chromosome

segregation or spindle pole body, or both, two were annotated as playing a role in mitotic cell division, and one annotated with meiosis. Furthermore, two complete protein complexes were contained in this module: (i) the Ndc80 protein complex, responsible for proper alignment and attachment of chromosomes and (ii) the DAM1-DUO1 protein complex, which translates the force generated by microtubule depolarisation into movement to facilitate chromosome segregation. The authors conclude that this module is responsible for the separation of chromosomes and cell division functions. Finally, only one component of the module, at the time of publication, had not been assigned a definite functional annotation in either CYGD or SGD. The authors therefore predicted the biological function of this unannotated gene to be chromosome segregation and gave further evidence to back up the validity of this prediction. In general, such predictions would then go on to be confirmed or rejected in biological experiments.

The study presented in this section is an example of an early application of community stucture detection to biological networks, which serves to demonstrate the ability of clustering techniques to identify biologically relevant modules. This analysis by Chen and Yuan paved the way for many subsequent similar analyses leading to community structure detection in biological networks becoming a well-established analytical tool.

2.2.4.2 Functional cartography: universal node role assignment

The high-level view that is provided by the community structure decompostition of a network can also be utlised to gain information regarding the nature of individual nodes. Indication of a node's role in the entire network can be inferred from its position in the network partition. This was demonstrated in [83] where Guimerà and Amaral defined a node classification scheme based on the assumption that nodes with the same role should have similar topological properties. A node's role is characterised according to two measures: within-community degree z-score and participation coefficient. The within-community degree z-score measures how well a node is connected with nodes in its own community and is defined as:

$$z_i = \frac{k_i - \overline{k_{s_i}}}{\sigma_{k_{s_i}}} \tag{2.6}$$

where k_i is the number of links node i makes with other nodes in its own module s_i, $\overline{k_{s_i}}$ is the average of k over all nodes in s_i, and $\sigma_{k_{s_i}}$ is the standard deviation of k in s_i.

The participation coefficient measures how uniformly a node's links are distributed among the communities of a partition and is defined as follows:

$$P_i = 1 - \sum_{s=1}^{N_M} \left(\frac{k_{is}}{k_i} \right)^2 \tag{2.7}$$

where N_M is the number of modules in the partition, k_{is} is the number of links node i has with nodes in module s and k_i is the degree of node i.

The node classification scheme can be summarised as follows. Based on the within-community degree z-score, nodes are classified as hubs if $z \geq 2.5$ and non-hubs if $z < 2.5$, i.e. hubs have a higher number of links with nodes in their own communities than non-hubs. Non-hubs are then classified into 4 roles: R1, ultra-peripheral nodes ($P \leq 0.05$), R2, peripheral nodes ($0.05 < P \leq 0.62$), R3, non-hub connector nodes ($0.62 < P \leq 0.8$) and R4, non-hub kinless nodes ($P > 0.8$). Hubs are also classified into 3 roles: R5, provincial hubs ($P \leq 0.3$), R6, connector hubs ($0.3 < P \leq 0.75$) and R7, global kinless hubs ($P > 0.75$).

The above node-role classification scheme was applied to the metabolic networks of 12 organisms: four bacteria, four eukaryotes and four archaea. The networks were constructed from KEGG pathways [96] where nodes i and j are connected if there is a chemical reaction where i is the substrate and j is the product, or vice versa, resulting in undirected interactions. Communities were detected using the stochastic optimisation technique, simulated annealing.

The nodes in each of the metabolic networks were then assigned roles according to the classification scheme defined above. For each network, a similar node role distribution was found, indicating that these universal roles apply generally to metabolism rather than being species specific. The authors then hypothesised that nodes with different roles are under different evolutionary constraints and pressures and that nodes with a more structurally relevant role are more essential and therefore would be more conserved across species.

To test this hypothesis, the rate of conservation of metabolites with different node categories was calculated across the different species and the following key results were found. R1 metabolites, ultra-peripheral nodes, had the highest loss rate. R1 nodes have a low within-module degree and a low participation coefficient indicating low structural relevance and therefore this loss rate appears reasonable. Conversely, R6 metabolites

(connector hubs) were found to have the lowest loss rate which is in agreement with their high within-module degree and their high participation coefficient. Similarly, R3 metabolites (non-hub connectors) were also found to be highly conserved. Although characterised by a low degree, their links are distributed among several modules and are therefore responsible for inter-module communication, similar to R6 nodes. Without R3 and R6 nodes, modules may be poorly connected or unconnected entirely. Therefore elimination of R3 and R6 nodes would have a large impact on global network fluxes. A highly structurally relevant role for R3 and R6 nodes explains their high conservation across species.

R5 nodes (provincial hubs) on the other hand were found to have a high loss rate. R5 nodes have a high degree with links mainly in their module. Therefore if removed, the pathways in which they involved may be backed up within the module and therefore their elimination would have a smaller impact that would probably be confined to their module. The structural differences between R5 nodes with R3 and R6 nodes is reflected in the difference in rate of conservation.

This application demonstrates the ability of community structure analysis to identify important and possibly essential nodes in metabolic networks. Where previously hub and non-hub classification has been shown to indicate essentiality in PPI networks [93], here Guimerà and Amaral propose a more sophisticated classification scheme to quantify structural relevance. The authors predict that similar results linking topologcal properties to functional importance would be found in PPI and other biological networks. Overall this application illustrates the potential of community structure detection to determine properties of individual nodes that would be otherwise invisible in a low level analysis of a network where only individual nodes would be considered.

2.2.4.3 Conservation of protein interactions across species

The previous application considered the conservation of node-roles across species, now focus turns to the conservations of interactions. Zinman et al. [230] investigate the conservation of protein interactions in an attempt to explain why a lower rate of conservation is observed in comparison to sequence data conservation. To do this, the link between conservation of protein interactions and the modular topology of PPI networks was explored.

A comparative analysis was performed using four model organisms: *S. cerevisiae*, *S. pombe*, *C. elegans* and *D. melanogaster*. For each organism co-expression, PPI, genetic interactions and sequence similarity data were retrieved. For each data type, species-specific networks were constructed using a probabilistic approach that assigns a score to each edge based on the likelihood of the two genes that it connects participating in the same biological process. Subsequently, integrated networks were constructed for each species by combining all four data sources such that an edge weight on the integrated network was the sum of the likelihood scores from the four different data source networks. It follows that if two genes, $g_{A,1}$ and $g_{A,2}$ are connected in the integrated network of species A and they have orthologs in the integrated network of species B, $g_{B,1}$ and $g_{B,2}$, that are also connected, then the edge $g_{A,1} - g_{A,2}$ is said to be directly conserved.

First a comparison between *S. cerevisiae* and *S. pombe*, two strains of yeast and therefore the two closest species in the study, was made. A baseline rate of conservation of interactions was calculated. 18.11% of interactions in the integrated network of *S. cerevisiae* were conserved in the integrated network of *S. pombe* and 22.18% conserved in the other direction. This conservation statistic is denoted as 18.11%/22.18%. Next the link between conservation of interactions and the community structure of the integrated networks was investigated. The Markov Clustering algorithm (MCL) [62], based on simulation of random walks was used to partition each of the networks into communities.

The interactions in each of the networks were then classified as "within-module-interactions" (WMI) and "between-module-interactions" (BMI) according to their position in the community structure. The definition of conservation was then extended such that if the interaction itself is not directly conserved but the two orthologs of the genes connected in species A are in the same module in species B, then this is known as "extended module" conservation.

It was found that the conservation rate of connected WMIs between *S. cerevisiae* and *S. pombe* was 46.54%/29.94%. For BMIs the rate was 16.17%/20.16%. The rate of conservation of WMIs was therefore higher than the baseline found earlier, and conversely, lower for the BMIs. Randomisation tests found the opposite results, indicating that the conservation of WMIs was not due to chance.

Extended module conservation was next explored. Again considering WMIs and BMIs separately, the conservation statistics were 49.66%/31.97% and 16.91%/20.79% respectively. This indicates two possible situations: (i) the interaction in species A does exist in species B but has not yet been experimentally derived or (ii) the direct interaction no

longer exists but the functional effect is retained as a result of the modular structure. Overall the results show that interactions within modules appear more highly conserved than those that connect modules. The above analysis was repeated for all pairwise species comparisons and similar results were found.

It is suggested that the rate of conservation of all interactions is decreased due to the effect of the low rate of conservation of between module interactions, thus offering a potential explanation for the difference seen between sequence similarity conservation and interaction conservation. Furthermore, it is suggested there there exists an analogous situation in sequence conservation. The overall sequence similarity between close species is lower than if only coding regions are considered. This indicates that coding regions are more likely to remain constant, whereas the intergenic regions have no evolutionary pressure to stay the same and therefore have a lower rate of sequence conservation. The same reasoning can be translated to the difference in the conservation of links within modules and between modules to deduce that there is possibly a strong selective pressure to maintain modular topology of the network. Furthermore, the results are said to shed light on the relationship between genes associated with very different phenotypes in close species. That is, although modules are conserved, the interactions between the modules may change at a higher rate and therefore if a module is involved in a function in one species, it can then become involved in a different function in another species due to the changing interactions between modules.

Overall the modular topology detected in the integrated network indicates both robustness, through the conservation of within module links and flexibility, by the changing interactions between modules. Both are properties which may confer advantages to an evolving species, supporting the previous discussion in Section 2.2.3. A study by Ryan et al. has also found similar results [178]. In combining the results in this section, with the previous results concerning the functional cartography of biological networks (Section 6.3.4), one could arrive at the hypothesis that interactions within modules are well conserved across species and although "connector" nodes generally conserve their role, their interaction partners can change. Again supporting the description of a modular topology being both flexible and robust simultaneously.

2.3 Community structure detection methods

The examples given in the previous section illustrate the value and versatility of community structure detection as an analytical tool in bioinformatics. Moreover these applications serve to demonstrate the need for the development of methodologies that can accurately uncover the modular structure of biological networks. Such methods have been studied since as far back as the 1970s under the guise of the graph partitioning problem in computer science. Furthermore, detecting communities via hierarchical clustering has for a long time been widely employed in sociology. More recently, in the last ten years or so, the problem has been adopted by the physics and applied mathematics communities. Consequently, a broad range of methodological approaches exists.

Despite the great variation in methodology, the main aim is common among all approaches: the detection of a set of densely connected communities, preferably at a low computational cost. This leads to the question of how to define a "good" partition of a network, and how to determine whether one partition is better or worse than another. The response to these questions has been to some extent formalised with the introduction of a measure known as modularity that quantifies how well defined the communities of a partition of a network are. It followed that community structure detection was transformed into an optimisation problem, where the larger the value of modularity achieved, the better the partition. Many methods have since been developed to cluster networks via modularity optimisation. However, modularity optimisation is known to be NP-hard [35] and therefore efficient algorithms to find global maximum modularity values in large networks are unlikely to exist. Consequently the majority of methods employ heuristic optimisation algorithms in order to yield as close to optimal as possible at a reasonable computational cost.

The method development that will be presented in this thesis is centred on modularity optimisation. Therefore a review of existing algorithms in this category is now presented. First an introduction to the modularity metric is given, followed by a summary of some of the more well known methods, as well as derivative approaches. Finally, despite its popularity, modularity optimisation has been shown to exhibit some limitations, which are acknowledged at the end of this section.

2.3.1 The Girvan-Newman betweenness algorithm and modularity

One of the most popular community structure detection algorithms was proposed by Girvan and Newman [74] as previously used in the Chen and Yuan study [43]. This method represented the start of the high level of interest in the problem by physicists. The Girvan-Newman algorithm (GN) is a divisive clustering algorithm that iteratively removes edges such that the underlying modules of a network are gradually revealed.

The betweenness of an edge is the number of shortest paths between any two vertices in the network that run along that edge, giving an indication of the importance of an edge in the context of the entire network. Within a module, edges are more densely connected than between modules by definition of community structure. Therefore within a module there are a larger number of paths available between two nodes than there are between two nodes in different modules. It follows that an edge that lies between communities will have a large number of shortest paths running along it due to the lack of alternative paths, which is reflected by a high betweenness value. Therefore removing edges with a high betweenness value should gradually disconnect modules from the rest of the network.

The GN algorithm is summarised as follows:

1. Calculate the betweenness score for all edges in the network.
2. Remove the edge with the highest betweenness and recalculate the betweenness for all edges affected by the removal of the edge.
3. Repeat until no edges remain in the network.

The process can be presented on a dendrogram where all nodes initially belong to one community and as the algorithm progresses and edges are removed new communities begin to form (Figure 2.7). A horizontal line on the dendrogram shows the structure of the partition at different levels of the algorithm (e.g. the cuts at lines A and B in Figure 2.7). The problem then lies in deciding which level of the dendrogram represents the best partition of the network.

To address this, Girvan and Newman introduced a measure known as modularity, Q, to evaluate how well-defined the communities of a partition of a network are [148]. A simple measure of quality is the fraction of all edges that lie within modules compared

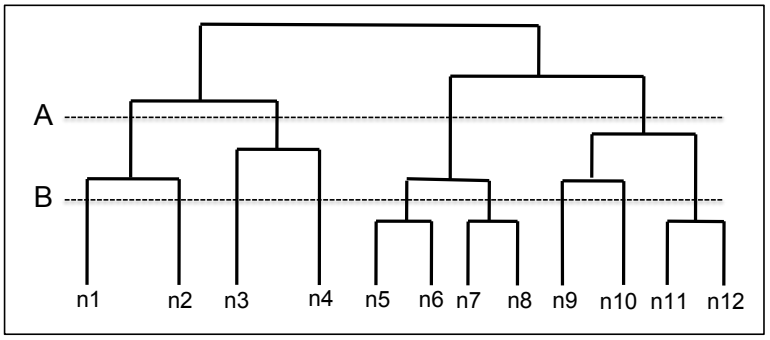

FIGURE 2.7: Example output dendrogram from the GN algorithm. The partition taken at cut A gives the following modules: $\{n1, n2\}$, $\{n3, n4\}$, $\{n5, n6, n7, n8\}$ and $\{n9, n10, n11, n12\}$. The communities at cut B are as follows: $\{n1\}$, $\{n2\}$, $\{n3\}$, $\{n4\}$, $\{n5, n6\}$, $\{n7, n8\}$, $\{n9\}$, $\{n10\}$ and $\{n11, n12\}$. Once the GN algorithm has terminated and the dendrogram constructed, the best partition is taken at the cut that corresponds to the highest value of modularity.

with the fraction that lie between modules, where a larger value of the former indicates a good partition. However, this measure is maximised when all nodes belong to one module which is not an interesting outcome. Therefore it was suggested that comparing the fraction of edges that lie within modules minus the expected value in a null model, i.e. a network with same degree distribution, but edges placed at random, would give a good measure of partition quality. A large value of Q indicates that the algorithm has found a good partition of the network into modules. It has been found that in general networks with strong community structure have Q between 0.3 and 0.7 [49]. The modularity measure is based on the assumption that a random network does not exhibit community structure. However, it has since been found that random networks can exhibit community structure [85, 169] and therefore a network is said to have community structure if it has a larger modularity that that of its corresponding randomised network.

The modularity of a network partition is defined as follows:

$$Q = \frac{1}{2L} \sum_{ij} \left(A_{ij} - \frac{d_i d_j}{2L} \right) \delta(C_i, C_j) \tag{2.8}$$

where A_{ij} is equal to 1 if there is a link between nodes i and j, d_i and d_j are the degrees of nodes i and j respectively, L is the total number of edges in the network and $\delta(C_i, C_j)$ is the Kronecker delta function equal to 1 if nodes i and j are in the same

module, 0 otherwise. The modularity metric was generalised to weighted networks in [145], discussed in more detail in Chapter 4.

Defining the following variables and summing over modules rather than edges, allows equation 2.8 to be transformed into the version that will be used in the rest of this thesis:

$$Y_{im} = \begin{cases} 1 & \text{if node } i \text{ belongs to module } m \\ 0 & \text{otherwise} \end{cases}$$

L_m number of links between nodes in module m

D_m sum of the degrees of the nodes in module m

More specifically,

$$L_m = \sum_{ij} \frac{A_{ij} Y_{im} Y_{jm}}{2} \quad \forall m \tag{2.9}$$

and

$$D_m = \sum_{i} d_i Y_{im} \quad \forall m \tag{2.10}$$

It follows that equation 2.8 can be re-written as:

$$Q = \frac{1}{2L} \sum_{ij} A_{ij} \delta(C_i, C_j) - \frac{1}{(2L)^2} \sum_{ij} d_i d_j \delta(C_i, C_j) \tag{2.11}$$

Replacing the Kronecker delta function with the Y_{im} variables and summing over m, equation 2.11 becomes:

$$Q = \frac{1}{2L} \sum_{m} \sum_{ij} A_{ij} Y_{im} Y_{jm} - \frac{1}{(2L)^2} \sum_{m} \sum_{ij} d_i d_j Y_{im} Y_{jm} \tag{2.12}$$

This can equally be written as:

$$Q = \frac{1}{2L} \sum_{m} \sum_{ij} A_{ij} Y_{im} Y_{jm} - \frac{1}{(2L)^2} \sum_{m} \left[\sum_{i} d_i Y_{im} \sum_{j} d_j Y_{jm} \right] \tag{2.13}$$

And subsequently:

$$Q = \frac{1}{2L} \sum_{m} \sum_{ij} A_{ij} Y_{im} Y_{jm} - \frac{1}{(2L)^2} \sum_{m} \left[\sum_{i} d_i Y_{im} \right]^2 \tag{2.14}$$

Finally, substituting in L_m and D_m gives:

$$Q = \sum_m \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right] \tag{2.15}$$

This version of the modularity measure will be used in the method development throughout this thesis.

According to the GN algorithm, in order to decide what level of the algorithm gives the best partition of the network, modularity is calculated at each stage on the dendrogram and the partition with the largest value is chosen as the final partition.

The GN algorithm has proven to be very popular and has been used as the basis of other clustering methods. For example, Holme et al. [91] modified the algorithm such that nodes are removed according to the value of node betweenness. More recently, a modification has been introduced by Sun et al. [193] where edges are not removed but their length is dynamically increased according to their betweenness. Unfortunately the GN algorithm has a fairly high computational cost, with the worst case run-time $\mathcal{O}(m^2 n)$ and for sparse graphs $\mathcal{O}(n^3)$ where n is the total number of vertices in the graph and m is the total number of edges. Consequently attempts to reduce the cost have since been proposed [43, 166, 198]. Most recently, Narayanan et al. [143] have introduced a step that identifies a near-optimal stopping point such that the full dendrogram does not have to be produced (i.e. the algorithm can terminate before all edges are removed) therefore reducing run time.

Following the introduction of modularity as a measure of the quality of community structure, modularity optimisation has been used as the basis for many clustering methods via various different optimisation techniques. A few of the most well known methods are discussed in the remainder of this section.

2.3.2 Greedy optimisation methods

Many greedy optimisation clustering methods exist. A greedy algorithm makes the locally optimal choice at each stage which results in fast algorithms, although sometimes at the expense of accuracy. However, the low computational cost is an advantage for clustering large networks and therefore a trade-off often has to be made.

One of the first greedy algorithms to optimise modularity was proposed by Newman [146]. This fast method uses an approximate optimisation strategy which can be applied to networks with up to a million vertices. The algorithm begins with all nodes belonging to individual communities. The adjacency matrix, A_{ij}, of a network with n nodes is an $n \times n$ matrix with entries equal to 1 if nodes i and j are connected, zero otherwise. The adjacency matrix involved in the greedy Newman algorithm is initially of size $n \times n$, but is updated as the procedure advances. Nodes are progressively grouped together based on the move that results in the largest increase in modularity. As nodes are joined together, they create clusters and in turn clusters are then combined, until finally all nodes belong to one module. The progression of the amalgamation of nodes/clusters is tracked on the adjacency matrix, updated at each stage. Rows and columns of the matrix refer to clusters instead of nodes as the algorithm progresses and the entries become the number of edges that join nodes belonging to two clusters divided by the total number of links in the network. The progress of the algorithm can be represented on a dendrogram, as with the GN algorithm, and the level with the highest modularity is the final partition.

Newmans greedy algorithm does not have a high computational demand ($\mathcal{O}((m+n)n)$ or $\mathcal{O}(n^2)$ for sparse graphs) since the change in modularity can be calculated in constant time and also the algorithm only considers adding nodes to a community where edges already exist and so this rules out many pairs, especially in a sparse graph. Despite the benfits of low computational cost, this often comes at the expense of low accuracy.

Clauset et al. [49] (CNM) went on to decrease the complexity of the greedy Newman method further by increasing the efficienty of the updating of the matrix A_{ij} by using data structures for sparse matrices, for example, max-heaps, which rearrange the data in the form of binary trees. The complexity becomes $\mathcal{O}(md \log n)$ ($\mathcal{O}(n \log^2 n)$ for sparse networks), where d is the depth of the dendrogram describing the clustering process, making the algorithm applicable to networks with up to 10^6 nodes. The CNM method is included in method comparisons in forthcoming chapters of this thesis.

Greedy optimisation of modularity can lead to large communities forming quickly at the expenses of smaller ones, which can result in inferior values of modularity [203]. Wakita and Tsurumi [203] proposed a modification of the Newman greedy method that merges nodes and clusters based on the increase in modularity times a consolodation ratio factor which peaks for communities of equal size, leading to a trade off between value of modularity achieved and balance between size of modules. This modification

lead to better values of modularity than those found by CNM and furthermore resulted in an increase in speed. Similarly, in order to avoid the formation of large communities, Schuetz and Caflisch [180] adapted the CNM algorithm such that more than one pair of nodes/clusters could be merged at each iteration. A refinement step where nodes could be moved between communities according to the increase in modularity was also added, leading to larger values of modarity although no increase in speed.

Blondel et al. [34] introduced a greedy iterative method, known as Louvain, that has the advantages of being computationally fast and accurate. Like with the Newman greedy method, the Louvain method combines nodes to form communities based on the moves that result in the maximum increase in modularity. Initially, all nodes are in individual mouldes and a sequential sweep over all vertices is carried out to determine which nodes should be merged according on the gain in modularity. After the first sweep, a new meta-network is formed, such that the communities that have formed become meta-nodes and the links between the communities are given the weight of the sum of the weights of the links that join the communities. The agglomerative algorithm is then applied to the new meta-network, resulting in another meta-network at a higher-level. This is repeated until no further improvement of modularity is possible. The output of the algorithm is a series of meta-networks, with the number of nodes decreasing after each pass, where the meta-nodes correspond to groups of nodes from the original network. The final pass gives the meta-network that corresponds to the best partition of the original network into communities. The complexity is $\mathcal{O}(m)$ therefore the method is very fast and is applicable on networks with up to 10^9 nodes, however it has also been shown to be accurate despite its greedy nature. The Louvain method is included in comparative analyses in forthcoming chapters.

The Louvain method has since been modified by De Meo et al. in [54] where nodes and edges are assigned weights based on a measure known as the k-path centrality which are then incorporated into the Louvain algorithm. Results found by the modified algorithm were found to be slightly better than those found by the original algorithm.

2.3.3 Simulated annealing

Guimerà et al. employed the stochastic optimisation technique of simulated annealing (SA) [83, 137] to detect communities. This is an iterative improvement method that finds the minimum of a cost function, C, which in the case of network partitioning, is

the modularity, Q, of the network, and since it requires to be maximised, $C = -Q$. The method begins with an arbitrary partition, p^0, of the network into between 2 and $n-1$ subsets, where n is the total number of nodes in the network. The nodes are randomly assigned to the subsets but with one subset left empty. The maximum value for modularity is found by transferring nodes from one subset to another with probability that depends on the global variable, τ, called the temperature. This transition probability is defined as follows:

$$q_{if} = \min(1, \exp(-(Q_i - Q_f)/\tau)) \tag{2.16}$$

where Q_i is the modularity of the initial partition and Q_f is the final modularity once the transition has been made. τ is monotonously decreased until the minimum of the cost function is found.

If p_k is a module in partition p^0, n_i a node in module p_k and another random module in partition p^0, p_l, are chosen. In the case where p_l is not empty, and there is no node n_j in p_l that is connected to n_i in p_k, n_i cannot be transferred to p_l, another n_i is chosen. If n_j does exist or p_l is empty, then the following Metropolis Acceptance analysis is carried out:

1. Calculate the modularity Q_i for the partition p^0.
2. Transfer n_i from p_k to p_l, which gives a new partition p^1.
3. Calculate the modularity, Q_f for the new partition p^1.
4. The new partition p^1 is accepted with the probability q_{if} above.
5. Steps 2 and 3 are repeated with the value of τ decreasing at each time step until no new configurations of the network are accepted.

SA finds accurate results on small to medium networks and is included in method comparisons in forthcoming chapters of this thesis. The complexity of the algorithm cannot be estimated, as it depends on the parameters chosen for the optimisation, not only on the graph size [72]. However, like the GN algorithm, it is slow (in fact, it is slower than GN) and so for large networks is inefficient.

The SA algorithm has been adapted to include an iterative k-means procedure by Liu and Liu in [126]. Furthermore, in [116], Lee et al. extend the SA algorithm to generate a global optimisation method by combining aspects of Monte Carlo with minimisation,

genetic algorithm and SA. Both methods reported better results than those found by SA alone. Nonetheless, the original SA algorithm is still commonly used in biological applications (e.g. see [207] and [222]).

2.3.4 Extremal optimisation

Duch and Arenas [60] proposed community structure detection via extremal optimisation (EO) with the aim of achieving values of modularity comparable with SA, but with reduced computational complexity. The heuristic method is based on the optimisation of local variables, expressing the contribution of each unit of the system to the global objective function to be optimised. In community structure detection, modularity, Q, is taken to be the global variable and the local variables are the contributions from the individual nodes to the sum of the modularity, which is then rescaled by the degree of the node and is called the fitness of the node. The algorithm can be summarised as follows:

1. Split the nodes randomly into two sets, with the same number of nodes in each set.
2. The node with the lowest fitness is moved from its set to the other set and the nodes involved have their fitness recalculated.
3. Repeat the process until the value of Q cannot be improved.
4. Delete the links between the two resulting communities and repeat the process on them separately.
5. This is continued until the value of Q cannot be improved.

The disadvantage of the EO algorithm is that the initial random partitioning of the vertices into two communities influences greatly the end result. Consequently, the nodes are ranked in order of their fitness resulting in the final partition of the network being much less sensitive to the initial conditions. EO has a computational complexity of $\mathcal{O}(n^2 \log n)$ and is faster than SA and GN but slower than the greedy algorithms [72]. In general it is thought that EO offers a good trade off between accuracy and speed [72].

The EO algorithm is generalised to weighted networks in [65]. Furthermore, the algorithm has been extended in [26] where the authors propose a method based on EO and

a random local search agent. It is shown that the results found by the extended version of the method improve on those found by EO, GN and the greedy Newman method.

2.3.5 Spectral optimisation methods

In [147], a method developed by Newman expresses modularity in terms of the spectral properties of the network. The algorithm computes the leading eigenvector of the modularity matrix and divides the vertices into two groups depending on the sign of the elements in this vector. The modularity of the network can be written as follows:

$$Q = \frac{1}{4L}\mathbf{s}^T\mathbf{B}\mathbf{s} \tag{2.17}$$

where $\mathbf{s}$ is a column vector such that $s_i = 1$ when node i is in group 1 and -1 if node i is placed in group 2. $\mathbf{B}$ is a real symmetric matrix such that the elements are as follows:

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2L} \tag{2.18}$$

where A_{ij} is equal to 1 if there is a link between nodes i and j, 0 otherwise, and k_i and k_j are the degrees of nodes i and j and L is the total number of edges in the network. Therefore, Q can then be written as:

$$Q = \frac{1}{4L}\sum_{i=1}^{n}(\mathbf{u}_i^T \cdot \mathbf{s})^2\beta_i \tag{2.19}$$

where the $\mathbf{u}_i$ are the normalised eigenvectors of matrix $\mathbf{B}$ and the β_i's are the corresponding eigenvalues. The eigenvalues are labeled such that β_1 is the largest, β_2 the second largest and so on.

Vector $\mathbf{s}$ is chosen such that the term in Q that involves β_1 is given as much weight as possible, therefore maximising the term involving $\mathbf{u}_1$, i.e. maximising the dot product $\mathbf{u}_1 \cdot \mathbf{s}$. Since the elements in the vector $\mathbf{s}$ have only the values $+1$ or -1, $\mathbf{s}$ is chosen such that $s_i = 1$ if the corresponding element in $\mathbf{u}_1$ is positive and $s_i = -1$ otherwise. This means that all the nodes whose corresponding elements in the leading eigenvector $\mathbf{u}_1$ that are positive are assigned to group 1 and the remaining nodes are assigned to group 2, giving the partition of the network into two communities.

In order to further divide the network, the spectral algorithm is applied to the two communities found by the first application. However, it is not sufficient to simply delete

the edges that link the two communities, therefore an extra contribution, ΔQ, is added to the modularity each time a group is further divided into two groups with the aim of maximising ΔQ and dividing the elements in a similar way to above. If no division exists that gives a positive value for ΔQ then the community should not be divided. The algorithm ends when the entire network has been divided into indivisible communities.

There are many advantages to this eigenvector-based method. Firstly, when the modularity matrix has no positive eigenvalue, the method is indicating that no good division of the network exists and thus there is no community structure. Similarly the signal that a module should no longer be divided is when there is no positive value for ΔQ. In addition, the elements of the leading eigenvector give a measure of how strong the membership of the corresponding node is to its group. For example a large positive element would indicate that the corresponding node plays a central role in the community and conversely a small element of the eigenvector would indicate that the node is a weaker member of the community and that it may not definitely belong to this community. This is demonstrated in the example of the karate club network [224], presented in Chapter 3, Section 3.3.2.1 where the largest elements of each eigenvector corresponded to the two ringleaders of the division of the karate club.

Newman's spectral method is fast and accurate. On a sparse graph the total computational cost is $\mathcal{O}(n^2 \log n)$, which is faster than the GN algorithm and although slower than the greedy Newman algorithm, it is shown to obtain more accurate results [147]. However it has been found that this method is less accurate when the number of communities is greater than two [72]. In [58], Du and Tan use the Newman spectral method in combination with a refinement step where nodes are moved between modules according to the increase in modularity to cluster a network of Chinese words.

Ruan and Zhang [176] use the spectral properties of the network Laplacian, L, and a local refinement step in order to optimise the network modularity in an algorithm known as QCUT, where the network Laplacian matrix is the diagonal matrix of node degrees minus the network adjacency matrix. It was observed that k-partitioning of the network found better partitions than recursive bi-partitioning [150] and therefore Ruan and Zhang developped QCUT by combining both to benefit from the accuracy of k-partitioning and the efficiency of bi-partitioning.

The algorithm involves two steps: partitioning and refinement. The partitioning stage splits the network into a partition with k modules by testing different values of k from

2 up to a user-defined value l (a small integer). The partitioning algorithm can be summarised as follows:

1. All nodes in the network are initially in a single module and initial modularity is set to zero, $Q_0 = 0$.
2. For $k = 2, ..., l$ carry out the following two steps:
   (i) Apply the NJW k-partitioning algorithm [150]: compute the k smallest generalised eigenvectors of the network Laplacian, L, and combine them to form the columns of matrix $\mathbf{Y}$. Normalise the rows of $\mathbf{Y}$ and apply the k-means algorithm in [61] to group the rows (i.e. nodes) into k clusters.
   (ii) Calculate the modularity of the whole network with the new partition, Q_k.
3. Choose the k that gives the best value of modularity, Q_k. If $Q_k > Q_0$, accept this as the new partition of the network and update Q_0 to Q_k.
4. For each of the k clusters in the new partition, repeat steps 2 and 3.
5. Repeat steps 2 to 4 until no further improvement of Q is possible.

The refinement stage follows using a local search strategy involving two types of operations: moving a vertex from one community to another (migration) and combining two communities to form a single one (merging). The process continues by returning to the partitioning stage to see if any of the communities affected by the refining stage can be further partitioned and then continues to alternate between the two stages until neither stage can improve the value of Q.

Since the Laplacian matrix is in general a sparse matrix this method uses less memory than the modularity matrix. As a result, QCUT is more efficient for larger networks than Newmans spectral algorithm. QCUT is included in method comparisons later in this thesis.

The spectral properties of the network Laplacian had been previously combined with hierarchical clustering as a network partitioning method by Donetti and Muoz [57]. Spectral methods have also been used recently by Nadakuditi and Newman [142] where the adjacency matrix, constructed by averaging over the ensemble of a stochastic block model of the network, and the modularity matrix are both investigated.

2.3.6 Mathematical programming

In [14], Agarwal and Kempe propose two mathematical programming formulations of modularity optimisation. The first is an integer linear programming (IP) model with variables x_{ij} equal to 0 if nodes i and j are in the same cluster and equal to 1 otherwise. Modularity is written as follows:

$$Q = \frac{1}{2L} \sum_{ij} B_{ij}(1 - x_{ij}) \tag{2.20}$$

where $\mathbf{B}$ is the modularity matrix defined in equation 2.18 and L is the total number of links in the network. Q is optimised subject to the following constraint:

$$x_{ik} \leq x_{ij} + x_{jk} \tag{2.21}$$

i.e. if nodes i and j are in the same module and nodes j and k are in the same module, then nodes i and k are also in the same module. Furthermore, an integrality constraint is imposed on the x_{ij} variables, i.e. $x_{ij} \in \{0, 1\}$. The above IP is NP-hard and therefore the integrality constraint is relaxed such that $0 \leq x_{ij} \leq 1$, then the problem can be solved in polynomial time. The new LP optimisation problem is solved using the commercial solver CPLEX [13] (see further on in this section for a description of CPLEX). However, there are now fractional solutions and therefore a rounding step is required. The rounding step consists of interpreting the x_{ij} variables as a distance measure between nodes i and j, such that the closer x_{ij} is to zero, the closer the nodes i and j are to each other and therefore are more likely to belong to the same community. The rounding stage is then followed by a post processing refinement step.

Next the authors propose a quadratic programming (QP) formulation of the problem which splits the network into two clusters and which can then be iteratively reapplied to each cluster to find further clusters and so on. Variables y_i are equal to $+1$ or -1 depending on which of the two clusters node i belongs to, similar to the Newman spectral method in Section 2.3.5. Modularity is then defined as follows:

$$Q = \frac{1}{4L} \sum_{ij} B_{ij}(1 + y_i y_j) \tag{2.22}$$

where again, $\mathbf{B}$ is as in equation 2.18 and L is the total number of links in the network. Q is maximised subject to $y_i^2 = 1$ for all i, which assures that the y_i variables are all equal to ± 1. Again the model is NP-complete and therefore the constraint that the y_i

variables must be integer is relaxed. This is done by replacing all y_i variables with vectors $\mathbf{y}_i$ and each $y_i y_j$ product with the dot product $\mathbf{y}_i \cdot \mathbf{y}_j$, resulting in a vector programming (VP) model which can be solved in polynomial time using semi-definite programming (SDP) (with solver CSDP). The rounding procedure then consists of choosing a random $(n-1)$-dimensional hyperplane (where n is the number of nodes in the network) to cut the hypersphere into two halves thus defining two clusters.

Both the IP and the QP formulations have a relatively high computational cost and are therefore inefficient on large networks. However, both methods find good results in comparison with the GN algorithm, the Newman spectral algorithm and EO on medium sized networks.

Neither of the above mathematical programming models can guarantee globally optimal solutions, however, in [219], Xu et al. formulate the modularity optimisation problem as a mixed integer quadratic programming (MIQP) model, which due to the convexity of the model can be solved to global optimality. This method, known as OptMod, is now described in detail as it is incorporated into the community stucture detection method development in chapters 3 and 7.

First the indices, sets and parameters associated with the mathematical model are listed below:

Indices

n, e	nodes
l	links
m, k	modules

Parameters

N	total number of nodes in the network
L	total number of links in the network
M	total number of modules in the partition
d_n	degree of node n

α minimum module size (if no bounds this can be equal to 1)

β maximum module size (if no bounds this can be equal to N)

ϵ maximum size difference between any pair of modules

Sets

S M most connected nodes

AM_n allowable modules for node n

ML_l allowable modules for link l

AV_m nodes allowed assignment to module m

B_n nodes with higher connectivity than node n

Binary variables

$$E_m = \begin{cases} 1 & \text{if module } m \text{ exists;} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{nm} = \begin{cases} 1 & \text{if node } n \text{ belongs to module } m; \\ 0 & \text{otherwise} \end{cases}$$

$$X_{lm} = \begin{cases} 1 & \text{if link } l \text{ belongs to module } m; \\ 0 & \text{otherwise} \end{cases}$$

Positive variables

L_m number of links between nodes in module m

D_m sum of the degrees of the nodes in module m

Modularity is defined as in equation 2.15:

$$Q = \sum_m \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right] \tag{2.23}$$

Q is maximised subject to several constraints described below.

First, L_m and D_m in equation 2.23 are defined as follows:

$$L_m = \sum_l X_{lm} \quad \forall m \tag{2.24}$$

$$D_m = \sum_n d_n Y_{nm} \quad \forall m \tag{2.25}$$

Where, as defined in Section 2.3.1, L_m and D_m are the number of links that lie fully within module m and the sum of the degrees of the nodes in module m respectively.

Next, OptMod detects a partition of disjoint modules of a network, therefore a constraint to ensure that each node belongs to only one module is required:

$$\sum_m Y_{nm} = 1 \quad \forall n \tag{2.26}$$

Link l belongs to module m only if both nodes associated with it, n and e, belong to module m. And nodes n and e can only belong to module m if module m is in both AM_n and AM_e. This condition can be defined in the following three constraints:

$$X_{lm} \leq Y_{nm} \quad \forall l = \{n, e\}, m \in ML_l \tag{2.27}$$

$$X_{lm} \leq X_{em} \quad \forall l = \{n, e\}, m \in ML_l \tag{2.28}$$

$$X_{lm} = 0 \quad \forall l, m \notin ML_l \tag{2.29}$$

Where $ML_l = AM_n \cap AM_e$.

The above is the basic optimisation model. However the authors include several other constraints to control the difference in size between modules and to tackle the problem of equivalent solutions therefore reducing computational cost. It is at this point that the flexibility of mathematical programming is illustrated.

A degeneracy constraint is proposed in order to assure that module m is only allowed to be in use if the previous module $m-1$ exists and therefore avoiding equivalent solutions:

$$E_m \leq E_{m-1} \quad \forall m = 2, ..., M \tag{2.30}$$

It follows that module m is not empty (i.e. $E_m = 1$) if the following two constraints hold at the same time:

$$\sum_l X_{lm} \geq \alpha E_m \quad \forall m \tag{2.31}$$

$$\sum_l X_{lm} \leq \beta E_m \quad \forall m \tag{2.32}$$

These two constraints also place bounds on the size of the modules, however if this is not desired, α can be equal to 1 and β equal to the total number of links in the network.

The authors next consider balancing constraints such that the difference in size between modules is at most ϵ. This is safeguarded by the following two constraints:

$$L_m - L_k \leq \epsilon + \beta(1 - E_k) \quad \forall m, k > m \tag{2.33}$$

$$L_k - L_m \leq \epsilon + \beta(1 - E_k) \quad \forall m, k > m \tag{2.34}$$

Again, this constraint is optional, depending on the users requirements in a specific clustering experiment.

Finally, symmetry breaking constraints are included to eliminate equivalent solutions, i.e. where the following two cases are generated as two separate solutions when they are in fact equivalent:
Solution 1: $m1 = \{n1, n2\}$, $m2 = \{n3, n4, n5\}$, $m3 = \{n6, n7\}$,
Solution 2: $m1 = \{n3, n4, n5\}$, $m2 = \{n6, n7\}$, $m3 = \{n1, n2\}$.
To avoid equivalent solutions produced by re-numbering modules, nodes can only be assigned to one of a particular set of modules, AM_n. To do this, the nodes are first ordered according to their connectivity. If $n1$ is the most connected, then it can only belong to module $m1$, if $n2$ is the second most connected then it can be assigned to modules $m1$ and $m2$ and similarly for the remaining $M - 2$ most connected nodes in set S. The remaining nodes that are not in set S can be assigned to any of the M modules. Consequently, constraint 2.26 becomes:

$$\sum_{m \in AM_n} Y_{nm} = 1 \quad \forall n \tag{2.35}$$

Furthermore, node n can only be allocated to module m (such that $m \in AM_n$) if at least one of the nodes with a higher connectivity than node n (members of the set B_n) which can be assigned to module $m - 1$ (also a member of the set AV_{m-1}) has been assigned

to module $m-1$. This is formulated mathematically as:

$$Y_{nm} \leq \sum_{e \in (B_n \cap AV_{m-1})} Y_{em-1} \quad \forall n \geq 3, m = 3, ..., |AM_n| \tag{2.36}$$

However, the authors of [219] found that the computational cost can be reduced by only considering the M most connected nodes (i.e. set S), therefore the constraint 2.36 becomes:

$$Y_{nm} \leq \sum_{e \in (B_n \cap AV_{m-1})} Y_{em-1} \quad \forall n \in S, n \geq 3, m = 3, ..., |AM_n| \tag{2.37}$$

Finally the resulting MIQP model, comprises a concave quadratic objective function that is maximised subject to a set of linear constraints and mixed binary/continuous optimisation variables. OptMod is formulated as follows:

Maximise:

$$Q = \sum_m \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right] \tag{2.38}$$

Subject to: Constraints (2.24, 2.25, 2.27-2.35, 2.37)

$$L_m,\, D_m \geq 0 \quad \forall m \tag{2.39}$$

$$E_m, X_{lm}, Y_{nm} \in \{0, 1\} \quad \forall n, m, l \tag{2.40}$$

As mentioned above, OptMod can guarantee globally optimal values of modularity due to the convexity of the model. In [219] the CPLEX mixed integer optimisation solver [13] is used to find globally optimal results. CPLEX, which was also mentioned above to solve the LP model in [14], is a tool that uses constraint programming to solve combinatorial optimisation problems such as Linear Programming (LP) and Mixed-Integer Programming (MIP) problems. The MIP algorithm is an implementation of a branch and bound search, where the MIP problem is relaxed by dropping the integrality conditions. Therefore in the case of OptMod, the binary variables are no longer binary, but become continuous variables and the MIQP problem becomes a quadratic programming (QP) problem, constituting the root of the tree. CPLEX then solves the QP using the Primal Simplex algorithm or the Dual Simplex algorithm and either:

(i) the solution satisfies the original integrality restrictions and therefore the solution obtained is optimal,

(ii) the QP problem is infeasible and therefore so is the original MIQP problem,

(iii) at least one of the integer variables in the solution is fractional in the QP solution.

If option (iii) above is true, then one or more of the fractional variables is chosen and the tree "branches" to create two or more subproblems. The prior solutions are then excluded but any feasible integer solutions are not eliminated. These new problems constitute the "nodes" of a branching tree. A QP problem is solved for each node created. CPLEX uses a combination of best-first search and "diving" or "plunging" to explore the nodes.

The authors of [219] report globally optimal solutions for networks of up to only 104 nodes are reported. Therefore, although OptMod is accurate, it is computationally expensive and therefore its applicability is limited. This limitation was addressed in [19] by Aloise et al. where an exact column-generation algorithm based on OptMod was proposed. Many linear programs are too large to consider all the variables explicitly therefore in column-generation algorithms only a subset of variables are considered when solving the problem. The original optimisation problem is split into two: the master problem and an auxilliary problem. The master problem is the original problem with a reduced set of variables. The auxilliary problem is a problem that is solved in order to find an entering column for the simplex algorithm (the method used by the CPLEX solver) where the columns correspond to all nonempty modules. Globally optimal partitions were found by the modified version of OptMod for networks of up to 512 nodes. Therefore, although an improvement on OptMod, this method still exhibits scalability limitations.

The same group have also incorporated the MIQP formulation of modularity optimisation into a locally optimal bi-partitioning method which is recursively appplied to split modules into two groups [36]. Each bi-partition of a module is solved optimally by the MIQP, but overall the recursive procedure is heuristic and therefore the global optimum value of modularity cannot be guaranteed. In a comparison with the exact optimisation of the column-generation method, the locally optimal method achieved sub-optimal results for all networks tested by both methods. However it is applicable to larger scale networks than the both of the exact methods described above therefore maybe more suitable for some clustering experiments.

Overall, mathematical programming offers a flexible modelling framework for developping clustering algorithms. As is illustrated by OptMod, multiple constraints can be

included to deal with different user needs, e.g. imposing bounds on module size. Furthermore, as has been seen from the various methods described above, it is possible to formulate the modularity optimisation problem in a variety of different models. In light of its amenable nature, mathematical programming will be used in this thesis to develop methods for tackling the community structure detection problem. More specifically, the starting point of this method development is the MIQP model, OptMod. The first aim being to develop a model that can increase the limited scalability of OptMod but that can retain to some extent the accuracy of the method. This will be explored in Chapter 3.

2.3.7 Summary

In this section a review of the most well-known modularity optimisation methods has been given. This is an area of research that has attracted a large amount of interest and as a result a complete review of all existing modularity optimisation methods is beyond the scope of this thesis. Here the aim was to illustrate the main approaches that have been taken to solve the modularity optimisation problem and in particular, describe the pioneering methods that have underpinned much of the method develpoment that has been carried out since the introduction of modularity. There are of course many more methods than those described in this chapter and new methods are appearing all the time, however the well-established methods, such as the GN method, CNM and the Newman spectral method are still commonly used in applications and are also often used as a means of benchmarking new methods.

Each method has its own advantages and disadvantages but when choosing a clustering method, the main consideration is the trade-off between accuracy and computational cost. For example, greedy methods such as the Newman greedy algorithm and CNM [49] are very fast and can partition large networks, however generally these methods are known to be less accurate. However, an exception to the rule is the Louvain method [34] which is faster than both of the other greedy methods but more accurate and in fact is accurate in comparison with non-greedy optimisation methods. On the other hand there are very accurate methods such as SA [83], however applicability is limited to medium sized networks. And even more extreme are the methods that find globally optimal values of modularity [219, 19] and therefore suffer even more strongly from scalability limitations. Consequently methods that are currently being developped aim to reduce the gap between accuracy and computational cost. However, overall, the choice

of method should be made based on the network being analysed and specific conditions of the clustering experiment.

Alternative methods to modularity optimisation also exist. These include:

- A method based on conductance, a measure of how good an individual community is, also known as the normalised cut metric by Leskovec et al. [118].
- Information theoretic approaches by Rosvall and Bergstrom [173, 174]
- Statistical inference/mixture model approaches by Newman and Leicht [149], Hastings [88], Xu et al. [220] and Zhao et al. [228].
- Methods based on random walks by Pons and Latapy [162], Enright (MCL) [62] and Cheng et al. [44].
- Pott's models by Li et al. [120] and Reichardt and Bornholdt [168].
- A method based on stockastic block modelling by Karrer and Newman [98].

This list includes just a few methods out of a very large number of existing non-modularity based methods and serves only to give an indication of the possible alternatives. However in this thesis various versions of the community structure detection problem are tackled and in most cases the approaches are based on modularity optimisation and therefore comparative analyses are made with other modularity optimisation methods. Consequently, related work focuses on modularity associated methodology. Any additional relevant work will be presented throughout the thesis where appropriate.

It must also be mentioned at this point that modularity optimisation is known to suffer from three limitations. First, as has been noted previously, modularity optimisation is NP-hard [35] and therefore developing accurate methods for large networks has proven to be one of the main challenges. Furthermore, concerns have been expressed regarding the degeneracy of the solutions found by modularity optimisation. The structure of partitions with modularity values close to that of the optimal partition can vary significantly which is problematic since most methods are heuristic and detect sub-optimal partitions [76]. Finally, it has been found that modularity optimisation suffers from a resolution limit, where small modules are not always detected [73]. This discovery put in doubt many of the methods that had previously been developed and consequently there

exists a large number of methods that attempt to overcome this property, e.g. [21, 121]. Alternatively, from a certain point of view this may not necessarily be a disadvantage. For example, Lewis et al. [119] argue that each level of community structure in the yeast PPI network is of biological interest. Therefore they conclude that being able to probe community structure at varying resolutions offers a deeper understanding of the modular organisation of biological systems.

The limitations of modularity optimisation and various means of overcoming them will be discussed in more detail in the final chapter. In the meantime it suffices to be aware that modularity optimisation may have its drawbacks but is it a widely accepted method and in fact is one of the most popular methods for community structure detection. Clustering methods based on modularity optimisation continue to be devloped as the benefits that they appear to confer often outweigh the disadvantges. Furthermore modularity can be easily extended to cluster more complex cases, e.g. networks with directed [117] or weighted [145] interactions. Overall, at the present time, modularity is considered to be a robust and useful measure that provides a natural, intuitive description of community structure for a wide range of real life networks. Its optimisation has proven its utility in many studies and in particular has found meaningful results in biological applications. It remains therefore a valid route to pursue in community structure detection method development.

2.4 Conclusions

This chapter has presented the necessary preliminaries for the work that will be described in the remainder of this thesis. First the main properties of complex networks and the various types of biological networks that they efficiently model were discussed. Focus turned to one property in particular: community structure. The community structure detection problem, its significance in biological systems and key bioinformatics applications were then described. Finally, modularity was introduced as a metric to quantify how well-defined the community structure of a network is, followed by a wide range of associated optimisation based clustering methods. All of the above underpins the work that will follow from this point onwards.

The applications of community structure detection to biological networks described in Section 2.2.4 emphasise the importance of clustering methods in bioinformatics and their capability to extract meaningful biological information. If the role of community

detection is to continue to feature in biological network analysis, then accurate methods to carry out the task are required. In particular, models that can effectively reproduce the complex and intricate relationships that exist in biological systems will be of great value in bioinformatics research.

The related work that has been discussed in this chapter has dealt with the standard community structure detection problem: the partitioning of an unweighted network into a series of disjoint communities. This problem statement represents the starting point of this thesis. However, the problem statement will continually evolve over the course of the forthcoming chapters, with several algorithms presented, each taking a step further towards a more realistic model of community structure. The first modification of the standard problem is the incorporation of weighted interactions. The clustering process is then extended in order to allow for overlapping communities where nodes can belong to more than one module. Finally, dynamics are integrated into the problem and the challenge of finding communities in networks that change over time is tackled. Each stage requires a modification or an extension of previous models and this is facilitated by taking a mathematical programming approach. The amenable nature of this modelling framework is conducive to the changing requirements of the various manifestations of the community structure detection problem featured in this thesis.

More precisely, the starting point of the clustering methodology development that will follow is the MIQP method, OptMod (Section 2.3.6). OptMod addresses the initial problem statement described above, however its guarantee of global optimality limits its applicability. The first aim is therefore to develop a mathematical programming method that detects disjoint partitions but that is applicable to larger scale networks than those accommodated by OptMod, while still retaining a high level of accuracy. This is now explored in Chapter 3.

Chapter 3

Detecting disjoint community structure in complex networks using integer optimisation

Community structure detection has been widely used as a means of revealing the underlying properties of complex networks. Since the adoption of modularity as a measure of how well defined the community structure of a network is, many methodologies for community structure detection based on its optimisation have been proposed. Due to the NP-hard nature of the optimisation problem, developing methods to efficiently detect satisfactory partitions of large networks has been particularly challenging. Therefore the search for faster and more accurate clustering methods is an important task. The modularity optimisation problem has previously been formulated as a mixed integer quadratic programming (MIQP) model. However, due to the guarantee of global optimality, the method suffers from scalability limitations. In this chapter the aim is therefore to extend this previous work in order to accurately cluster larger scale networks. To this end, the MIQP model is incorporated into a novel two-stage mathematical programming approach where an initial partition of a network is first detected and is then improved through an iterative optimisation procedure. A comparative analysis shows that despite no guarantee of optimality, the procedure finds globally optimal solutions on networks of up to approximately 500 nodes and furthermore outperforms all other methods tested when applied to larger networks. This chapter not only increases the scalability of the

existing mathematical programming approach, but also illustrates the ease in which such a framework can be extended and modified to meet changing experimental requirements.

3.1 Introduction

Topological properties of networks, and more specifically community structure, have proven important in revealing the underlying organisation of complex networks. The arrival of modularity [148], a metric to express the quality of community structure proved to be an important breakthrough in the community structure detection problem. Modularity measures the difference between the fraction of links within communities and the expected fraction of the same value when links are allocated randomly on a network with the same degree distribution. Modularity is discussed in more detail in Chapter 2, Section 2.3.

The modularity metric has transformed the community structure identification problem into an optimisation task where community structures can be determined by maximising the network modularity through various optimisation techniques. A review of existing methods is given in Chapter 2, Section 2.3, illustrating the diversity of approaches taken to tackle the optimisation problem. As modularity optimisation is NP-hard [35], efficient algorithms to find global maximum modularity values in large networks are unlikely to exist. Therefore, most approaches employ heuristics that aim at finding near-optimal solutions with modest computational cost. Methods that do achieve globally optimal solutions have however been proposed [19, 219]. In particular, in [219], Xu et al. formulated modularity optimisation as a mixed integer quadratic programming (MIQP) model known as OptMod. Details of the algorithm are given in Chapter 2, Section 2.3.6. The method was reported to find solutions in small networks with up to 104 nodes therefore demonstrating limited applicability. In this chapter, the aim is to adopt the mathematical programming framework of the MIQP model and develop the methodology such that accurately clustering larger scale networks is achievable.

This chapter is structured as follows. In the next section, a novel two-stage mathematical programming approach, known as iMod, is proposed. Stage 1 comprises a mixed integer non-linear programming (MINLP) formulation of modularity optimisation that detects an initial partition of the network. The MINLP model is non-convex and as such does not guarantee global optimality. Stage 2 therefore incorporates the MIQP model, OptMod in an iterative improvement procedure. The performance of iMod is

assessed via a comparative analysis with several methods from the literature on synthetic and real life networks. The methodology presented in this chapter improves on existing techniques for community structure identification so as to increase efficiency and applicability. Also demonstrated in this chapter is the flexibility offered by using mathematical programming as a modelling framework.

3.2 A two-stage mathematical programming model for detecting disjoint communities in complex networks

The work in this chapter has been a collaborative effort with Dr. Gang Xu. Initial model development and implementation was carried out by Dr. Gang Xu. My contribution to this chapter has been the generation of synthetic networks, collection of network data, determining appropriate methods for comparison and finding their implementations, running all network clustering experiments and the analysis of the results. Specifically, the work in Section 3.2 is the contribution of Dr. Gang Xu and all other sections in this Chapter can be ascribed to myself.

The approach presented in this section is a two-stage modularity optimisation procedure, known as iMod, which uncovers disjoint community structure in unweighted and undirected complex networks. In Stage 1, an MINLP model (MINLP_Mod) is formulated to obtain a locally optimal initial solution. In Stage 2, the solution obtained in the first stage is improved through an iterative optimisation procedure employing the MIQP model, OptMod [219]. A schematic representation of the entire module detection strategy, iMod, encompassing Stages 1 and 2 of the computational procedure is shown in Figure 3.1. Overall, the iMod approach is intended to extend the application of the mathematical programming framework to larger size networks than previously attainable by OptMod alone.

3.2.1 Stage 1: detecting the initial partition

As defined in Chapter 2, given an unweighted, undirected network with N nodes and L links, the modularity metric, Q, of a network partitioned into M communities is represented as:

$$Q = \sum_{m} \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right] \tag{3.1}$$

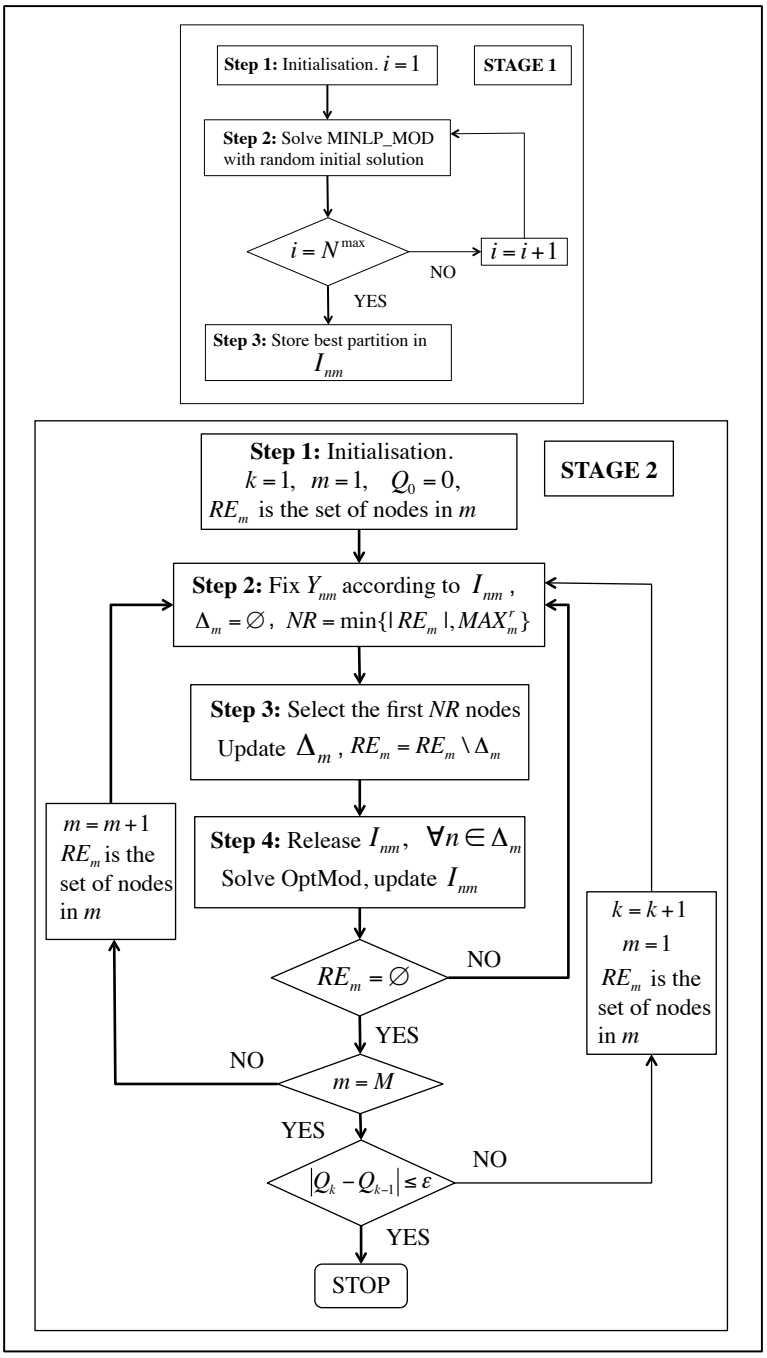

FIGURE 3.1: Flowchart of the iMod algorithm. Full details of Stages 1 and 2 are described in the text.

where L_m denotes the number of links in module m and D_m is the degree of all nodes in module m. Here, modularity optimisation is formulated as an MINLP model with the objective function as defined in equation 3.1. The indices and parameters associated with MINLP_Mod are presented below:

Indices

n, e nodes

m modules

i iterations

Sets

CN_m the set of nodes, e, that are connected to node n

Parameters

d_n degree of node n

L total number of links in the network

N^{max} the number of times the MINLP is solved in one run of MINLP_Mod

Binary variables

$$Y_{nm} = \begin{cases} 1 & \text{if node } n \text{ belongs to module } m \\ 0 & \text{otherwise} \end{cases}$$

I_{nm} for storing the node-module allocation of the partition with the best value of modularity in the N^{max} solves of the MINLP

Continuous variables

L_m number of links between nodes in module m

D_m sum of the degrees of the nodes in module m

Modularity is maximised subject to a number of constraints described below. First, since all modules are disjoint, each node can belong to only one module:

$$\sum_{m} Y_{nm} = 1 \quad \forall n \tag{3.2}$$

where Y_{nm} is a binary variable taking the value of 1 if node n is allocated to module m; 0 otherwise. As previously defined, D_m is equal to the sum of the degrees of nodes allocated to module m:

$$D_m = \sum_{n} d_n Y_{nm} \quad \forall m \tag{3.3}$$

where d_n is degree of node n. Finally, a link will be allocated to module m only when both nodes associated with it are also in module m. Therefore, the total number of links in module m, L_m, is defined by the following nonlinear equality:

$$L_m = \sum_{n} \sum_{\substack{e>n \\ e \in CN_n}} Y_{nm} Y_{em} \quad \forall m \tag{3.4}$$

where CN_n is the set of nodes e connected to node n. Overall, the resulting MINLP model (MINLP_Mod) is formulated as:
Maximise:

$$Q = \sum_{m} \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right] \tag{3.5}$$

subject to: Constraints (3.2-3.4)

$$L_m,\, D_m \geq 0 \quad \forall m \tag{3.6}$$

$$Y_{nm} \in \{0, 1\} \quad \forall n, m \tag{3.7}$$

Since global optimality of non-convex MINLP models cannot be guaranteed, MINLP_Mod is solved for a given number of times, N^{max}, each with a different random initial solution. The node-module allocation corresponding to the largest value of modularity of the N^{max} runs is stored in the variable I_{nm}. Using $N^{max} = 100$ provides a good representation of solution space and is used to generate the computational results in Section 3.3.

3.2.2 Stage 2: iterative improvement of the initial partition

Having selected the solution with maximum modularity from the Stage 1 (stored in the variable I_{nm}), the partition can now be improved through the iterative fixing and releasing of nodes. The idea is to solve a reduced version of the MIQP formulation of modularity optimisation method, OptMod, as previously proposed in [219]. For full details of the algorithm, please refer to Chapter 2, Section 2.3.6. The Y_{nm} variables are fixed according to the partition I_{nm} and sets of nodes are released iteratively, making them free to be re-allocated to a different module in subsequent solves of the MIQP. The releasing of a relatively small fraction of nodes reduces the number of variables, resulting in a more easily solvable model. Here, some additional indices and parameters associated with Stage 2 are introduced below:

Indices

k the iteration number of each complete round of the solution improvement procedure

Parameters

M total number of modules in the initial solution from Stage 1

NR total number of nodes released

MAX_m^r the maximum number of nodes to be released

$Aver_m$ the average degree of module m, excluding links made with nodes outside module m

U user defined parameter involved in the calculation of MAX_m^r

Sets

RE_m the nodes in module m, where module m contains the nodes currently released

Δ_m the nodes in module m that are released

Variables

Q_k the value of modularity after one major iteration of the improvement procedure, where each module has taken a turn to have its nodes released

Stage 2 begins by fixing the Y_{nm} binary variables according to the partition I_{nm}. An initial module, m, is chosen, where the set of nodes in the module is denoted by RE_m. A subset of nodes in RE_m is released (or un-fixed), denoted as Δ_m, where the size of Δ_m is NR. To avoid releasing too many nodes so that the reduced OptMod model is still reasonably easy to solve, the maximum number of released nodes for module m, MAX_m^r is set to:

$$MAX_m^r = \frac{U}{Aver_m} \tag{3.8}$$

where $Aver_m$ denotes the average degree in module m without considering the inter-module links and U is a user-defined parameter. Here, $U = 200$, which was shown to provide satisfactory results for all examples studied. As a result, the actual number of released nodes, NR, is the smaller value between MAX_m^r and the number of remaining nodes to be released in RE_m (i.e. $NR = \min\{|RE_m|, MAX_m^r\}$). In other words, if the number of nodes in module m is greater than MAX_m^r, the first NR nodes, Δ_m, will be released and the reduced MIQP solved. I_{nm} is updated and RE_m becomes $RE_m \setminus \Delta_m$. If the updated RE_m is still greater than MAX_m^r, a further set of nodes of size NR is released, otherwise all remaining nodes are released. The reduced MIQP is solved once again and I_{nm} and RE_m are updated accordingly. This is repeated until all nodes in the module have been released at one point and the procedure moves on to the next module. The set of NR nodes released in modules where the total number of nodes is greater than NR, is determined as follows. First the nodes are sorted with non-decreasing indices and then higher priority is assigned to nodes with smaller indices.

The above scheme is applied sequentially to each module in the partition, completing one round of the solution improvement iteration, k, with the final network modularity value, Q_k. The same strategy starts again, retaining the processing order of the modules, until no improvement of the modularity value is reported for two successive major iterations.

Here, the Zachary karate network example described in Section 2.2.2 is once again employed in order to demonstrate the above procedure. First $|RE_m|$, $Aver_m$, MAX_m^r and NR are defined for module 1 (corresponding to the pink nodes in Figure 3.2), where here

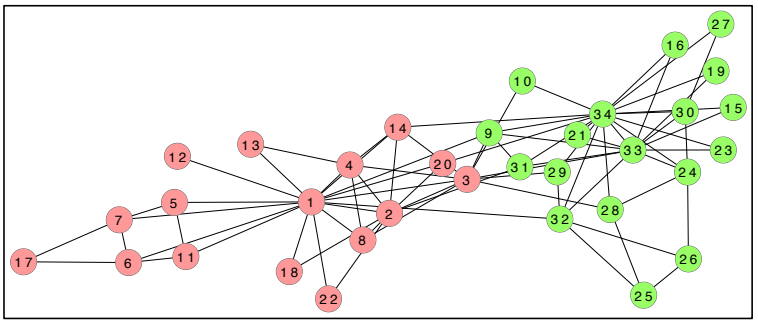

FIGURE 3.2: Karate club network two-module partition. The pink nodes correspond to module 1 as described in the example of the iMod iterative procedure.

$U = 50$ for the purpose of illustration and initially the possible set of nodes to be released for module 1 is, $RE_{m1} = \{1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 17, 18, 20, 22\}$. Therefore:

$$|RE_{m1}| = 16$$

$$Aver_{m1} = 4.13$$

$$MAX^r_{m1} = \frac{U}{Aver_{m1}} = \frac{50}{4.13} = 12.12$$

$$NR = \min\{|RE_{m1}|,\ MAX^r_{m1}\} = \min\{16, 12\}$$

Where MAX^r_{m1} is rounded down to 12.

In other words, module 1 in the known partition of the karate network comprises 16 nodes, which have an average intra-module degree of 4.13. Consequently, the maximum number of nodes that can be released from module 1 is 12 and therefore on this first iteration, 12 nodes are released as this is less than the total number of nodes in the module. If the nodes are to be released in order of non-decreasing indices, then the set of the first 12 nodes to be released is: $\Delta_{m1} = \{1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14\}$.

At this point, the MIQP model, OptMod, is solved such that all other node-module allocations are fixed and only nodes in Δ_{m1} have the possibility of being assigned new module memberships. The partition is then updated according to the solution found in this iteration and the corresponding value of modularity is recorded. However, there remain nodes in module 1 that have not yet been released in this round of the improvement procedure. It follows that once again the node-module allocations are fixed and RE_{m1} and consequently, NR are recalculated. The remaining nodes to be released is the set

$RE_{m1} \setminus \Delta_{m1} = \{17, 18, 20, 22\}$. Therefore $NR = \min\{|RE_{m1}|, MAX^r_{m1}\} = \min\{4, 12\}$ and as such, all four remaining nodes are released and OptMod once again solved.

Again the partition is updated and the corresponding value of Q recorded. Module 1 is now finished with for this round of the improvement procedure, and the above process is repeated for module 2. Once both modules 1 and 2 have been subject to the node fixing and releasing scheme, a complete round of the improvement procedure has been completed. The procedure is then repeated and continues to be repeated until no improvement of Q is reported for two successive complete rounds.

Comparing the full MIQP model, OptMod, to the reduced MIQP model implemented in iMod, the latter strategy involves fewer variables and constraints and is therefore more efficient in the case of larger size networks. Computational results are reported in the next section.

3.3 Results

In this section, the application of iMod is demonstrated through a comparative analysis with other modularity optimisation module detection methodologies on a series of synthetic and real network examples. All implementations of iMod are performed in GAMS (General Algebraic Modelling System) [172] and mathematical models (MINLP and MIQP) are solved using SBB [1] (see Section 4.4.2.3 for a description) and CPLEX [13] (see Section 2.3.6 for a description) mixed integer optimisation solvers respectively, with computational limit of 100000 seconds. All experiments were run remotely on a bioinformatics Sun Fire X4450 Server running 16 Xeon(R) E7340 processors at 2.4GHz and 32GB of PC2-5300 667 MHz ECC fully buffered DDR2 memory. The server runs CentOS Linux release 5.8 OS.

3.3.1 Synthetic networks

The performance of iMod is evaluated on a large number of artificial networks with known community structure generated according to the experimental design by Newman and Girvan in [148]. These synthetic networks comprise 128 nodes and are partitioned into four communities of 32 nodes with degree equal to 16. Additionally, networks with degree equal to 5 are also considered, as this represents a more realistic estimate of

the average node degree in the real networks discussed in the forthcoming section (see Tables 3.1 and 3.4). The idea is that the mixing parameter, μ (the fraction of all links in a particular module that end outside the module), is varied such that as μ increases, the modules of the true community structure become less well defined and therefore less easily detected. Testing for a mixing parameter greater than 0.5 is not deemed necessary, as it contradicts the definition of community structure, where more intra-community links than inter-community links should exist. Here the ability of iMod to extract known community structures is assessed and compared with that of the Louvain method [34] (see Chapter 2, Section 2.3.2 for details of the Louvain method).

A measure of similarity is required to ascertain how close the detected partition is to the true partition. In [107], Lancichinetti and Fortunato propose the normalised mutual information as a measure of similarity between two partitions, which ranges from 0 (for dissimilar) to 1 for identical community structures. This measure is taken from information theory and intuitively is how much information is needed in order to infer one partition from the other. The mutual information between two partitions is defined as follows:

$$I(X,Y) = \frac{2(H(X) - H(X|Y))}{H(X) + H(Y)} \tag{3.9}$$

where X and Y are the random variables associated with two partitions, and

$$H(X) = -\sum_{x} P(x) \log P(x) \tag{3.10}$$

is the marginal entropy, and

$$H(X|Y) = -\sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(y)} \tag{3.11}$$

is the conditional entropy. For each cluster $x \in X$, $P(x) = \frac{n_x}{n}$, where n_x is the number of nodes in module x and n is the total number of nodes in the network, is the probability that a node will belong to module x. The joint probability distribution of a node belonging to module $x \in X$ and module $y \in Y$ is $P(x,y) = \frac{|n_x \cap n_y|}{n}$. When two partitions are identical, the conditional entropy is equal to zero, that is, $I(X,Y)$ is equal to 1. An implementation of the mutual information measure was downloaded from [69].

100 synthetic networks were generated for each mixing parameter from 0.1 to 0.5 (at intervals of 0.05) using software downloaded from [70]. Each network is partitioned with both iMod and Louvain (implementation for Louvain downloaded from [81]) and the

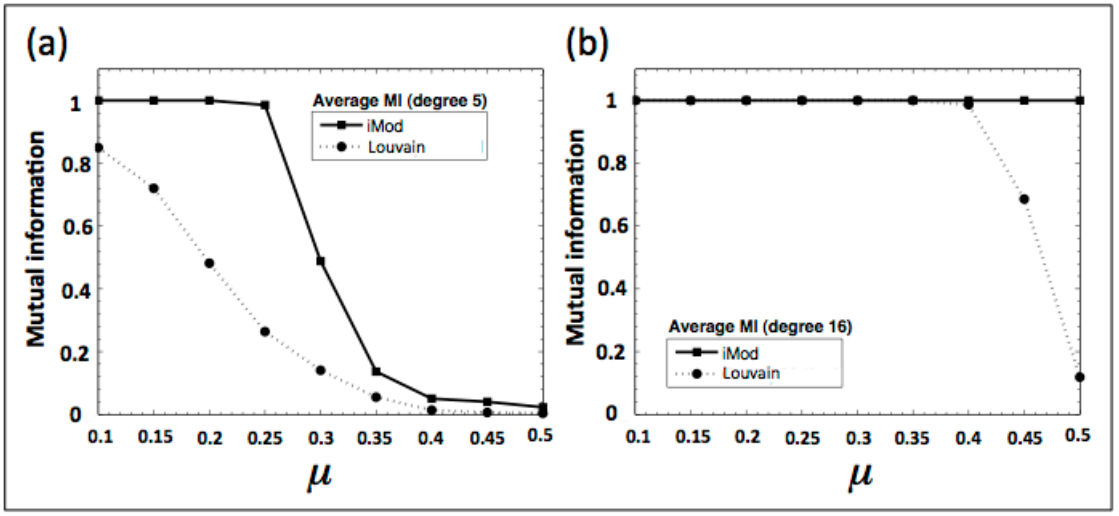

FIGURE 3.3: Benchmarking of module detection performance with iMod and Louvain. Synthetic network examples (128 nodes, 4 modules) were generated with node degrees of 5 and 16 in (a) and (b) respectively. For each mixing parameter, μ, 100 networks were assessed. The agreement of modules detected with the known community structure was expressed via the mutual information measure. Consistently better performance was noted for iMod.

average mutual information over the 100 networks is calculated. Figure 3.3 (a) and 3.3 (b) report the mutual information plotted against the mixing parameter for the synthetic networks to illustrate how close these methods are to revealing the known community structure. Overall, iMod performs better than Louvain in the examples tested. For node degree equal to 16, iMod and Louvain manage to retrieve the exact partition for all values of up to 0.35. Thereafter, iMod outperforms Louvain by continuing to extract the exact partition whereas Louvain's performance declines rapidly. In the case of degree equal to 5, iMod achieves higher similarity to the known structure than Louvain for all values of μ.

3.3.2 Real life networks

The synthetic networks tested above help to demonstrate the accuracy of iMod, however, the networks remain fairly unrealistic. Therefore, the performance of iMod is now tested on several real life networks.

3.3.2.1 Exact optimisation

OptMod finds the global optimum of the modularity maximisation problem and in [219] solutions for networks up to 104 nodes are reported. Due to the NP-hardness of modularity optimisation, exact solutions remain restricted to small networks. However, in [19], OptMod is extended using column generation methods for linear integer programming, giving another method guaranteeing optimality (denoted here as the Exact method). In [19], Exact is reported to detect globally optimal community structures for networks with up to 512 nodes. These network sizes and densities are summarised in Table 3.1. Density is defined as the actual number of edges divided by the possible number of edges, $\frac{2L}{N(N-1)}$, where L is the total number of edges and N is the total number of nodes in the network. The networks are briefly described below.

Overall, 11 examples were used from [19] with varying sizes, inspired from social or biological relationships, many being well-studied cases in network analysis and related algorithm development. First is the Zachary karate club network [224], the well-studied network often used in benchmarking tests for community structure detection method development, described in Section 2.2.2. Despite the network's known community structure comprising two modules, representing the two newly formed karate clubs resulting from as dispute between the club president and instructor, it has been shown that the partition of the network with the optimal value of modularity comprises 4 modules [219], which is the partition that modularity optimisation methods seek to achieve.

The remaining networks are:

- Lusseau's dolphin dataset describing communications between dolphins during a field study in Doubtful Sound New Zealand [130, 131].
- Victor Hugo's Les Miserables dataset, compiled by Knuth [101], describing interactions between characters in the novel of the same name.
- The main connected components of two datasets on classes and relationships from a software project related to graph drawing (A00 main and A01 main) [2].
- The p53 protein protein interaction (PPI) network [53].
- Krebs' political book dataset modelling the co-purchasing of books about US politics from Amazon (Polbooks)[102].

Network	**Nodes**	**Links**	**Density**
Karate	34	78	0.1390
Dolphin	62	159	0.0841
Les Miserables	77	254	0.0868
A00_main	83	125	0.0367
Protein p53	104	226	0.0422
Polbooks	105	441	0.0808
American football	115	613	0.0935
A01_main	249	635	0.0206
USAir97	332	2126	0.0387
NetScience_main	379	914	0.0128
Electronic circuit	512	819	0.0063

TABLE 3.1: Summary of the test networks used in [19].

- A network modelling the schedule of football games between teams in an American university league [74].
- A network describing the flight schedule between US airports in 1997 [3].
- The main connected component of a dataset on a coauthorship network of scientists working on network theory compiled by Newman [147].
- A network describing electronic circuits [20].

iMod is applied to each of the above networks. For each community detection experiment on a particular network, iMod was run 10 times, resetting the seed for the random number generator each time. iMod is compared with several other modularity optimisation algorithms from the literature. The column generation extension of OptMod [19] (Exact), is included as a benchmarking method due to the values of modularity reported in the article being the global optimal values for the test networks in Table 3.2. iMod is also compared against two greedy agglomerative methods, CNM [49] and Louvain [34], a spectral partitioning method, QCUT [176] and the stochastic optimisation method, simulated annealing (SA) [83, 137]. For more details of each methods see Chapter 2, Section 2.3. The results are shown in Table 3.2, where the values of modularity found by Exact are those reported in [19]. The software for CNM, Louvain, QCUT and SA, was downloaded from [48, 81, 175, 105] respectively and as before all experiments were run remotely on a bioinformatics Sun Fire X4450 Server. For each clustering experiment, each of the above methods was run once.

Network	iMod		Exact		CNM		Louvain		QCUT		SA	
	Best Q	**M**	**Best Q**	**M**	**Best Q**	**M**	**Best Q**	**M**	**Best Q**	**M**	**Best Q**	**M**
Karate	**0.4198**	4	**0.4198**	4	0.3807	3	0.4188	4	0.4188	4	**0.4198**	4
Dolphin	**0.5285**	5	**0.5285**	5	0.4955	4	0.5185	5	0.5175	5	0.5268	4
Les Miserables	**0.5600**	6	**0.5600**	6	0.5006	5	0.5556	6	**0.5600**	6	**0.5600**	6
A00_main	**0.5309**	9	**0.5309**	9	0.5239	7	0.5294	9	0.5281	7	0.5253	6
Protein p53	**0.5351**	7	**0.5351**	7	0.5205	8	0.5274	7	0.5219	8	0.5299	6
Polbooks	**0.5272**	5	**0.5272**	5	0.5020	4	0.5205	4	0.5208	4	**0.5272**	5
American football	**0.6046**	10	**0.6046**	10	0.5773	7	**0.6046**	10	**0.6046**	10	0.6044	10
A01_main	**0.6329**	14	**0.6329**	14	0.5991	12	0.6270	12	0.6257	13	0.6253	9
USAir97	**0.3682**	6	**0.3682**	6	0.3204	7	0.3508	7	0.3665	6	0.3656	6
NetScience_main	**0.8486**	19	**0.8486**	19	0.8383	19	0.8475	19	0.8467	15	0.8454	16
Electronic circuit	**0.8194**	12	**0.8194**	12	0.8056	12	0.7967	15	0.8161	13	0.8039	12

TABLE 3.2: Computational results comparing the performance of modularity optimisation methodologies across several network examples. Globally optimal values of modularity (as found by Exact in [19]) are denoted in bold.

For each network, Table 3.2 shows the best value of modularity found by iMod over the 10 runs and the number of modules in the partition corresponding to the best value of modularity. Results show that for each network tested, iMod detects the partition with the global optimal value of modularity as has been previously found by the Exact method. Louvain, QCUT and SA also achieve globally optimal solutions for two or three of the networks but CNM finds only sub optimal solutions. Overall, iMod outperforms CNM, Louvain, QCUT and SA by achieving globally optimal solutions for the complete set of test networks. The optimal partitions detected by iMod for four of the networks are illustrated in Figure 3.4 where each module is represented by a different colour. Due to the small scale of the networks, these visualisations can show clearly the members of each module and give a good idea of the modular structures.

A breakdown of the iMod results is presented in Table 3.3. Stage 1 modularity gives the best and median value of modularity detected by MINLP_Mod alone (before applying the improvement stage) over the 10 runs. Stage 2 modularity shows the best and median final modularity detected by iMod over the 10 runs. It can be seen that for some of the smaller networks, MINLP_Mod alone can achieve optimal solutions, however as network size increases, the iterative MIQP stage does indeed improve the final modularity values. For example, MINLP_Mod alone does not find the globally optimal solution for A00 main, but running the full iMod procedure does.

The CPU time of the algorithm is also broken down where the given value for each stage in Table 3.3 is the average CPU over the 10 runs. In terms of CPU time, MINLP_Mod

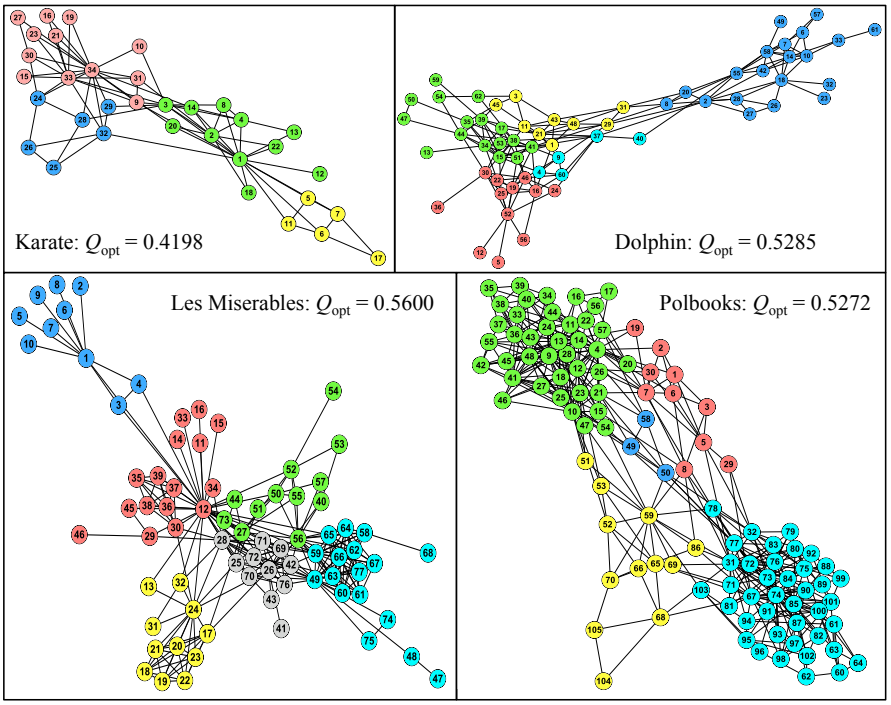

FIGURE 3.4: Visualisations of the optimal partitions detected by iMod for four of the benchmark networks. Images generated in Cytoscape [187].

increases exponentially with the size of network. However, this is not the case for the improvement stage due to the nature of the iterative procedure. For example, large differences in CPU can be seen for the American football and USAir97 networks which are not related to the size of the networks. Stopping time for Stage 2 depends on how quickly the criteria that two subsequent runs of the improvement stage achieve negligibly different values of modularity. Furthermore, the running time for Stage 2 is also dependent on the number of modules in the partition found by Stage 1, which does not necessarily depend on network size. Therefore, the increase in modularity offered by Stage 2 often comes at a large computational cost. For example, in the case of the American football network, a network of only 115 nodes, Stage 1 has an average CPU of 19.77 seconds, but the average CPU for Stage 2 is 1312.80. The large increase in computational cost represents the difference in achieving a sub optimal solution or the globally optimal solution. This is a clear example of the dilemma between accuracy and computational cost that is a major challenge in the development of community structure detection methods.

	Stage 1		Stage 2		CPU (seconds)		
Network	**Best**	**Median**	**Best**	**Median**	**Stage 1**	**Stage 2**	**Total**
Karate	0.4198	0.4198	0.4198	0.4198	0.28	0.08	0.35
Dolphin	0.5285	0.5276	0.5285	0.5285	0.90	1.62	2.52
Les Miserables	0.5600	0.5560	0.5600	0.5600	2.42	2.33	4.75
A00_main	0.5237	0.5132	0.5309	0.5280	3.99	0.77	4.76
Protein p53	0.5314	0.5208	0.5351	0.5335	2.12	2.34	4.46
Polbooks	0.5272	0.5270	0.5272	0.5272	4.33	306.42	310.75
American football	0.5845	0.5732	0.6046	0.6045	19.77	1312.80	1332.86
A01_main	0.6099	0.5998	0.6329	0.6326	25.75	322.12	347.88
USAir97	0.3609	0.3591	0.3682	0.3675	40.58	63.62	104.20
NetScience_main	0.7789	0.7648	0.8486	0.8478	62.31	237.59	299.90
Electronic circuit	0.7127	0.7008	0.8194	0.8187	299.07	208.64	507.71

TABLE 3.3: Breakdown of the iMod results into Stage 1 modularity (MINLP_Mod) and Stage 2 modularity (final modularity after the improvement stage) for networks in Table 3.1. Stage 1 gives the best and median value of modularity over the 10 runs and similarly for Stage 2 modularity. CPU is the average CPU in seconds over the 10 runs for each stage of the algorithm and the combination of both stages.

3.3.2.2 Locally optimal partitions of larger networks

The comparison with the Exact method limits the analysis to the networks reported in [19], therefore here iMod is tested on several other networks chosen because of their biological nature or due to their size. These include four biological networks:

- The transcriptional network of the bacterium *Escherichia coli* [183].
- The transcriptional network of the yeast *Saccharomyces cerevisiae* [139].
- The network of metabolic reactions of the nematode *Caenorhabditis elegans* [94].
- The main connected component of the rat protein protein interaction network downloaded from BioGrid [191].

Additionally, a network the email communications in a university is included, giving an example of a much larger network [84]. Network properties are summarised in Table 3.4. Once again, iMod is compared against widely used modularity optimisation approaches: CNM, Louvain, QCUT and SA. Results are shown in Table 3.5, where again the best value of modularity detected by iMod across the 10 runs and corresponding number of modules are reported.

Network	**Nodes**	**Links**	**Density**
E. coli	418	519	0.0060
C. elegans	453	2025	0.0198
S. cerevisiae	688	1079	0.0046
Rat PPI	811	946	0.0029
Email	1133	5451	0.0085

TABLE 3.4: Summary of the additional networks used in the method comparison in Section 3.3.2.2.

	iMod		**CNM**		**Louvain**		**QCUT**		**SA**	
Network	**Best Q**	**M**	**Best Q**	**M**	**Best Q**	**M**	**Best Q**	**M**	**Best Q**	**M**
E. coli	**0.7815**	40	0.7785	40	0.7790	41	0.7761	39	0.7783	40
C. elegans	**0.4525**	9	0.4061	9	0.4407	9	0.4361	9	0.4375	9
S. cerevisiae	**0.7746**	27	0.7596	27	0.7641	26	0.7647	27	0.5612	27
Rat PPI	**0.8445**	22	0.8436	22	0.8429	19	0.8425	18	0	1
Email	**0.5789**	8	0.5148	12	0.5426	11	0.5762	12	0.5740	9

TABLE 3.5: Computational results of the networks in Table 3.4 where the best modularity achieved across all methodologies is denoted in bold.

As seen in the previous section, consistently better performance is noted for iMod throughout all examples studied. Simulated annealing performs well for the *E. coli*, *C. elegans*, and email networks, however running the method on the *S. cerevisiae* and rat PPI networks produces unusual results. For the *S. cerevisiae* network, SA finds 27 modules, in agreement with the other methods but with a much lower value of modularity, however for the rat PPI network, SA places all nodes into one module, resulting in zero modularity. The reasons for these results are not clear at this stage.

As before the iMod results are now broken down into Stage 1 and Stage 2 modularity and CPU, shown in Table 3.6. Since the networks studied in this section are generally larger than those in Table 3.1, in all cases better modularity values are achieved when the full algorithm is run, demonstrating the value of running the second stage of the iMod procedure. Similar trends to the previous set of networks are shown in terms of CPU time. It is concluded therefore that, as is well-known in clustering methods, if necessary a trade-off must be made between accuracy and computational cost according to experiment specific requirements. In the context of this thesis however, the aim is to develop a clustering method that detects high values of modularity and that is competitive with other modularity optimisation methods. As evidenced through all of the above synthetic and real network examples, this has been achieved by iMod.

	Stage 1		Stage 2		CPU (seconds)		
Network	**Best**	**Median**	**Best**	**Median**	**Stage 1**	**Stage 2**	**Total**
E. coli	0.7202	0.7114	0.7815	0.7815	178.14	2144.51	2322.65
C. elegans	0.4290	0.4231	0.4525	0.4507	149.01	3407.93	3556.94
S. cerevisiae	0.7287	0.7131	0.7746	0.7745	320.47	9407.60	9728.07
Rat PPI	0.7856	0.7786	0.8445	0.8431	122.62	209.67	332.29
Email	0.5498	0.5484	0.5789	0.5726	330.99	3780.63	4111.62

TABLE 3.6: Breakdown of the iMod results into Stage 1 modularity (MINLP_Mod) and Stage 2 modularity (final modularity after the improvement stage) for networks in Table 3.4. Stage 1 gives the best and median value of modularity over the 10 runs and similarly for Stage 2 modularity. CPU is the average CPU seconds over the 10 runs for each stage of the algorithm and the combination of both stages.

Finally, Figure 3.5 shows the full email network visualised in Cytoscape [187]. This is a good example of the hair ball type network, where, due to a large number of nodes and links it is difficult to gain any information about the network from its visualisation alone. Consequently, visualising the partition in the same way as the networks in Figure 3.4 is not reasonable. Therefore, Figure 3.6, shows the meta-network of the partition of the email network found by iMod. Here, the meta-nodes clearly show the number of modules and their relative sizes. Furthermore, the thickness of the meta-links connecting the meta-nodes is correlated to the number of links between the nodes in the corresponding modules. The high level view of the network showing the community structure gives a more informative representation of the network topology than the low level view given by the full network. This illustrates that uncovering the community structure of a network is a good first step leading to a better overall understanding of large datasets.

3.3.2.3 Additional results reported in the literature

In addition to the methods tested above, several other modularity optimisation methods have also reported modularity values for some of the networks tested in this section. The results shown in Table 3.7 are reported in [147] for the edge betweenness method of Newman and Girvan (GN) [148], Newman's spectral method (Spectral) [147] and the extremal optimisation method of Duch and Arenas (EO) [60]. Once again, in all cases, iMod achieves higher values of modularity than those reported by these three methods. This serves as more evidence of the better quality performance of iMod that has been reported throughout this results section.

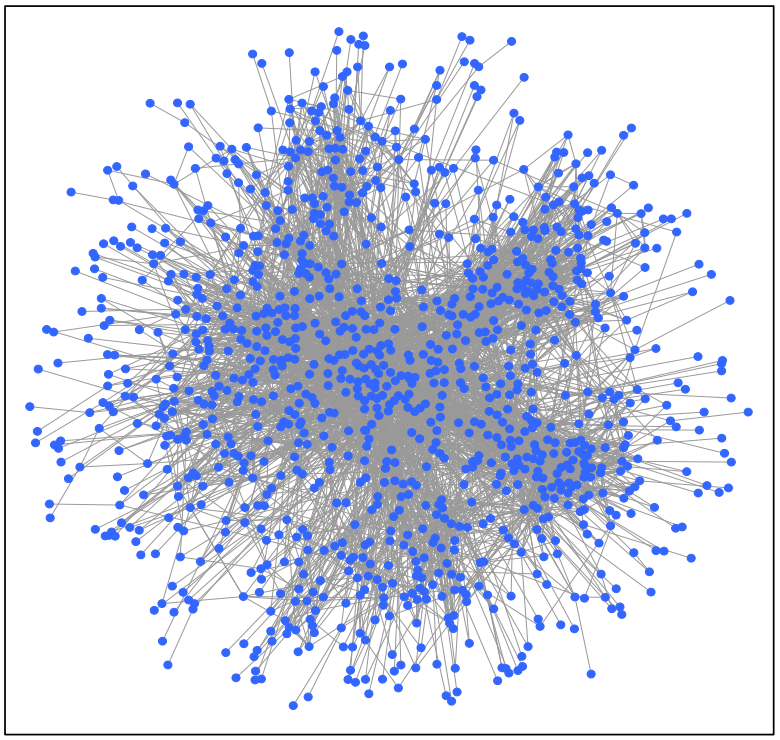

FIGURE 3.5: Visualisation of the email network in Cytoscape [187]. Good example of the hair ball network, where nodes and links are so numerous it is difficult to gain much information about the network structure.

Network	**iMod**	**GN**	**Spectral**	**EO**
Karate	**0.420**	0.401	0.419	0.419
Dolphin	**0.529**	0.520	-	-
Les Miserables	**0.560**	0.540	-	-
E. coli	**0.782**	-	0.766	-
C. elegans	**0.453**	0.403	0.435	0.434
S. cerevisiae	**0.775**	-	0.759	-
Email	**0.579**	0.532	0.572	0.574

TABLE 3.7: Additional results for the GN, Spectral and EO methods reported in [147].

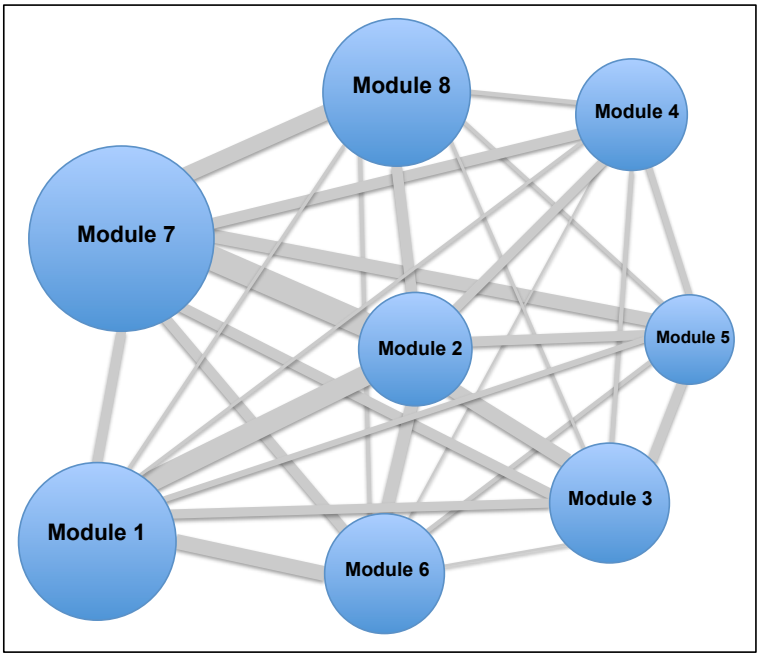

FIGURE 3.6: Meta-network of the partition of the email network found by iMod where each meta-node represents a module in the partition and each link is an aggregation of the links connecting the nodes in the modules. The size of each meta-node is correlated to the number of nodes in the corresponding module and similarly, the thickness of the meta-links is correlated to the number of links between each module.

3.4 Discussion and conclusions

Despite the large number of modularity optimisation algorithms that have been developed for community structure detection, the approach still suffers from the limitation of NP-hardness. As a result, modularity optimisation tends to lead to either accurate methods that are only applicable to small to medium sized networks, or heuristic methods that compromise accuracy in favour of scalability. Therefore the development of efficient clustering methods remains challenging.

In this chapter, a two-stage mathematical programming approach to network partitioning, known as iMod, was proposed. The procedure extends the previous work in which modularity optimisation is formulated as an MIQP model, guaranteeing global optimal solutions (OptMod) [219]. The high computational cost limits OptMod to small networks and therefore the aim here was to develop the methodology such that the accurate

clustering of larger networks is attainable. First, modularity optimisation was formulated as an MINLP model, which could be solved to find a good initial partition of the network. This initial partition is then improved through the iterative solving of the reduced MIQP model. It was shown through a number of network examples that the inclusion of the second stage does indeed improve the solutions detected by the MINLP. In some cases, the increase in modularity came at the expense of a much higher computational cost due to the nature of the improvement stage of the algorithm. Furthermore, the CPU time for Stage 2 in relation to network size appears unstable and, even for a network of only 115 nodes Stage 2 incurred a disproportionately high computational cost. As discussed in the results section, experiment specific requirements will dictate if a trade-off must be made between accuracy and speed. However, in this chapter, the priority lies with achieving higher values of modularity. Future work will consider means of decreasing computational cost while retaining quality of solution. This may be achievable through the use of alternative solvers (as is investigated briefly in the next chapter) or though the inclusion of symmetry breaking constraints in the MINLP in Stage 1 of iMod, similar to those that appear in OptMod.

A comprehensive comparison was carried out between iMod and several other modularity optimisation methods from the literature. First iMod and four other methods were compared with an exact optimisation approach previously shown to detect globally optimal solutions for networks with up to 512 nodes. For each network tested, iMod detected the globally optimal partition, which in most cases was not achieved by the other methods. iMod was then applied to several other networks with unknown optimal partitions and once again achieved higher modularity values than all other methods tested on networks with up to 1133 nodes. Therefore iMod not only demonstrates the desired improved scalability with respect to OptMod, but also has the ability to achieve globally optimal solutions in some cases and performs better than several other existing modularity optimisation methods.

Although iMod performs well in the comparative analysis, improvements may lie in considering alternative procedures for selecting the nodes to be released and the order in which they are released and additionally investigating the effect that changing the processing order of the modules would have on the results. Investigating such aspects will be included in future work.

Finally, the development of a new clustering algorithm through combining existing and novel optimisation models in order to increase scalability illustrates the capacity of the

mathematical programming framework to extend and adapt in order to meet new criteria. It is noted that iMod addresses a basic form of the community structure detection problem and that more informative models may be required to better represent real life systems, for example, through the accommodation of weighted interactions, overlapping modules and network dynamics. The flexibility of the modelling framework will be further explored throughout this thesis as variations of the community structure detection problem are tackled. As a starting point, Chapter 4 now addresses the problem of clustering weighted complex networks.

Chapter 4

Detecting disjoint community structure in weighted complex networks

Complex networks offer an efficient framework for modelling real life systems and the analysis of such networks can elucidate underlying properties of a systems organisation. Consequently, it is desirable to create more realistic models of complex systems in order to extract more meaningful information from their network representations. A first step in this direction is accommodating strengths of interactions that indicate the importance of an association between two entities in a system. In this chapter, with a view to developing a more informative clustering method, the mixed integer non-linear programming model (MINLP_Mod), which constitutes Stage 1 of the iMod procedure described in Chapter 3, is generalised to weighted networks. Furthermore, the number of times that the MINLP is solved in a single clustering experiment is increased and it is proposed that the approach acts as a stand-alone method, without the improvement stage featured in iMod. The performance of the method is tested on a series of synthetic and real networks against competitor approaches from the literature. Results show that the MINLP formulation of modularity optimisation can find sufficiently accurate results in order to compete with the other modularity optimisation methods tested, in particular on weighted networks. However in a comparison with iMod, it is shown that in some cases, the improvement stage can still add value to final solutions. Finally it is suggested that the MINLP model has the potential to act as a template for tackling various

aspects of the community structure detection problem and several alternative objective functions are suggested as replacements for modularity. Overall this chapter represents a step forward in the development of more accurate and informative clustering models, therefore contributing towards one of the main goals of this thesis.

4.1 Introduction

Complex networks can arise from different real world situations such as social interactions, the Internet and biological systems. In its most basic form, a complex network is unweighted (binary), i.e. if a link between two nodes exists, the adjacency of these nodes is given a value of one, otherwise zero. However, richer network representations can be implemented, with weights assigned to each link denoting the strength or confidence of an interaction. For example, in a collaboration network, weights may correspond to the number of papers co-authored by scientists indicating the strength of their collaboration. Similarly, in a network generated from gene expression data weights can reflect the degree of correlation between the expression profiles of two genes. In both examples, homogenous links may not sufficiently capture the true nature of the system being modelled.

In many early implementations of community structure detection, weights were often ignored either to simplify the problem or due to algorithm constraints, often discarding critical information about the nature of the network. However, the importance of taking weights into account has since been demonstrated [65, 125]. In [65], unweighted and weighted versions of the same network are clustered and their community structures compared, showing that the presence of weighted interactions affects the resultant network partition. Moreover, in [125], an iterative scoring method is proposed for assigning weights to protein protein interactions in order to assess the reliability of the interaction. It is found that the scoring method can reduce the impact of random noise on the ability of clustering algorithms to detect known protein complexes, illustrating that including weights can lead to more accurate solutions. The development of clustering algorithms that can accommodate weighted interactions is therefore important in the search of more realistic solution procedures. Consequently, the modularity metric has been generalised to weighted networks [145] with the result that weighted interactions can now be easily incorporated into many existing clustering algorithms. It follows that the modularity

optimisation methods developed in this thesis thus far can also be adapted in order to cluster weighted networks.

In Chapter 3 the task of detecting disjoint communities in unweighted networks was approached via a two-step procedure, iMod, comprising the detection of an initial solution followed by an iterative improvement process. In this chapter the next stage in the development of this methodology is presented. Here, the MINLP model (MINLP_Mod) comprised in Stage 1 of iMod is generalised to weighted networks and is now known as WeiMod to reflect this adjustment. Furthermore, it was seen in Chapter 3 that Stage 2 of the iMod procedure often incurred a disproportionately high computational cost, which was not directly linked to network size. Therefore a second aim of this chapter is to determine whether the MINLP formulation of modularity optimisation can be a sufficiently accurate method in its own right, without the inclusion of the improvement procedure. As such, the MINLP is now solved 1000 times in one run of WeiMod, as opposed to 100 times in MINLP_Mod, in an attempt to compensate for the loss of the iterative improvement stage.

This chapter is structured as follows. First, background information and related work are presented, including the definition of modularity applicable to weighted networks. Next, the MINLP model (WeiMod) that addresses the detection of disjoint community structure in weighted networks is outlined. The applicability of WeiMod is demonstrated through a comparative analysis with other modularity optimisation approaches from the literature on a series of real and synthetic networks. The performance of WeiMod is further assessed in a comparison with iMod to determine whether the MINLP method may still benefit from the improvement step despite the increase in the number of times the MINLP is solved in a single clustering experiment. Additionally, the results found by WeiMod are further analysed to consider the significance of the modularity values detected and the choice of commercial MINLP solver. Finally, it is suggested that WeiMod can offer a general template model for tackling various aspects of the community structure detection problem, with example alternative objective functions discussed. Overall, the model presented in this chapter extends previous work through the incorporation of provisions for weighted interactions therefore representing a first step in the direction of a more realistic description of complex systems.

4.2 Background and related work

Modularity was generalised to weighted networks by Newman in [145], and is defined as follows:

$$Q = \frac{1}{2L} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2L} \right) \delta(c_i, c_j) \tag{4.1}$$

where A_{ij} is the weight of the edge between nodes i and j, L is the sum of the weights of all the edges in the network and k_i is the strength of node i, i.e. the weighted degree of node i defined as the sum of the weights of the nodes' connections with other nodes in the network. As in the original version of modularity, $\delta(c_i, c_j)$ is the Kronecker delta function, where the value is 1 if nodes i and j are in the same module; 0 otherwise. It follows that many clustering algorithms developed to partition unweighted networks can be easily modified to partition weighted networks. It is noted that some methods developed after Newman's weighted modularity definition do not distinguish between the two cases as the application to both weighted and unweighted networks is implicit. Therefore the overview of modularity optimisation methods given in Chapter 2, Section 2.3 is relevant to both weighted and unweighted network analysis and should be referred to for related work corresponding to this chapter. Here, a few brief examples are given to illustrate the link between clustering unweighted and weighted networks. For more details of the methods, see Chapter 2.

First, the GN algorithm [74], the original method to adopt modularity as a measure of the quality of a partition is adapted to cluster weighted networks in [145]. The GN method is a divisive method based on removing edges according to their betweenness value. In brief, the transformation is as follows. First the weights of the network are ignored and the edge betweenness is calculated as normal. Each edge betweenness is then divided by the weight of the corresponding edge. This is derived from the idea of mapping the weighted network onto a multi-edge unweighted network such that if an edge between two nodes has weight 3, then this can be equally represented by three edges of unit weight between the two nodes. The algorithm proceeds as normal and the best partition is found by detecting the stage that has the highest modularity according to equation 4.1. This generalised version of the algorithm does not greatly increase the computational cost of the original GN algorithm.

The CNM method is a greedy agglomerative clustering algorithm [49]. The method begins with all nodes belonging to individual clusters. The adjacency matrix, A_{ij}, that

represents this network at this stage is of size $n \times n$ for a network with n nodes. In the case of unweighted networks, A_{ij}, is equal to 1 if nodes i and j are connected and for weighted networks, the entry is the weight of the edge between the two nodes. Nodes are progressively grouped together based on the move that results in the largest increase in modularity where for weighted networks, the original modularity definition is simply replaced by equation 4.1. As nodes are joined together, they create clusters and in turn clusters are then combined, until finally all nodes belong to one module. The progression of the amalgamation of nodes/clusters is tracked on the adjacency matrix, updated at each stage. Rows and columns of the matrix refer to clusters instead of nodes as the algorithm progresses and the entries become the sum of the weights of the edges that join nodes belonging to two clusters divided by the sum of the weights of all links in the network. This modification does not increase the running time.

The QCUT algorithm by Ruan and Zhang [176], is based on the network Laplacian matrix, i.e. the diagonal matrix of node degrees minus the network adjacency matrix. In the case of weighted networks, node strength is used in the diagonal matrix and edge weights in the adjacency matrix. No additional computational cost is incurred.

The GN, CNM and QCUT algorithms were originally described as clustering methods for unweighted networks. The Louvain method [34] on the other hand was directly proposed as a clustering method for weighted networks, using the definition of modularity given in equation 4.1 and therefore needs no modification. This is partly due to the fact that the nature of the algorithm requires it to construct weighted meta-networks to be re-partitioned iteratively, therefore requiring the ability to cluster weighted networks.

In general modularity optimisation methods, even if not explicitly stated, can be easily modified to incorporate weighted interactions. In summary, algorithms can be generalised to cluster weighted networks as follows: (i) binary values in adjacency matrices are replaces with edge weights, (ii) node degree is replaced with node strength and (iii) the total number of edges in a network is replaced with the sum of the weights of all the edges in a network. In many cases the transformation results in no extra computational cost.

It follows that the mathematical programming models presented in Chapter 3 can also be extended to weighted networks, as is demonstrated in the following section.

4.3 A mathematical programming model for detecting disjoint community structure in weighted and unweighted networks

In Chapter 3, a two-stage clustering procedure for unweighted networks was proposed. Stage 1, MINLP_Mod, finds a locally optimal value of modularity which is then ameliorated in Stage 2 via an iterative improvement stage incorporating the globally optimal method, OptMod [219]. In this section, Stage 1 is generalised to detect disjoint communities in weighted networks, with the algorithm now known as WeiMod.

In addition to allowing for weighted interactions, in a further step towards developing a more informative model, WeiMod also take loops into account, i.e. self-interactions, as illustrated in Figure 4.1. For example, accounting for loops allows a more precise representation of the situation where a transcription factor regulates its own transcription, i.e. auto-regulation, in gene regulatory networks.

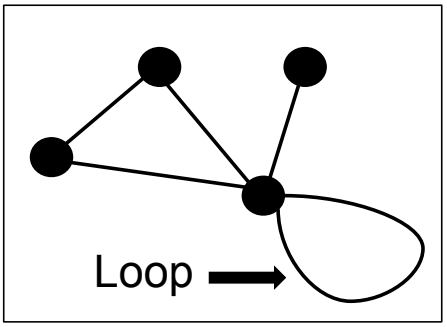

FIGURE 4.1: Example of a network containing a loop, i.e. a self-interaction.

Many of the indices and parameters associated with WeiMod are the same as those involved in MINLP_Mod, all are given below:

Indices

n, e nodes

m modules

Parameters

β_{ne} weight of the link between nodes n and e

α_n weight of the edge node n makes with itself i.e. a loop

d_n strength (weighted degree) of node n

L sum of the weights of all edges in the network

Binary variables

$$Y_{nm} = \begin{cases} 1 & \text{if node } n \text{ belongs to module } m \\ 0 & \text{otherwise} \end{cases}$$

Continuous variables

L_m sum of weights of all links among nodes within module m

D_m sum of strengths of the nodes in module m

As before, if modularity is summed across modules rather than nodes, the modularity measure for weighted networks in equation 4.1 can be re-written as follows:

$$Q = \sum_m \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right] \tag{4.2}$$

Modularity is maximised subject to constraints described below. First, all modules are disjoint, therefore each node in the network can only belong to one module:

$$\sum_m Y_{nm} = 1 \quad \forall n \tag{4.3}$$

where Y_{nm} is a binary variable, equal to 1 if node n is allocated to module m and zero otherwise. The total degree of module m, D_m, is calculated by:

$$D_m = \sum_n d_n Y_{nm} \quad \forall m \tag{4.4}$$

where the strength of a node n is defined as $d_n = 2\alpha_n + \sum_e \beta_{ne}$. This is the first significant difference between MINLP_Mod and WeiMod. First, node strength is now employed instead of node degree but additionally the inclusion of the α_n parameter

accounts for any loops in the network. The weight of the loop at node n (α_n) is counted twice in the node strength to account for the fact that in the sum of the degrees/strengths of all nodes in the network, each link is counted twice due to its association with two nodes. Therefore each loop edge must also be counted twice.

The final constraint is that for an edge to belong to a module, both nodes associated with it must belong to that module. It follows that the total sum of the weights of all links in module m, L_m, is defined as the following nonlinear equality:

$$L_m = \sum_n \alpha_n Y_{nm} + \sum_{n,e} \beta_{ne} Y_{nm} Y_{em} \quad \forall m \tag{4.5}$$

where node e is connected to node n. Again this differs in two ways from the equivalent constraint in MINLP_Mod. First the equation includes the parameter for the weight of the edge between nodes n and e, β_{ne}. Second, any loops of the nodes in module m are also accounted for by the inclusion of the α_n parameter.

The resulting MINLP model (WeiMod) comprises a non-linear objective function which is maximised subject to a set of linear and non-linear constraints with a combination of integer and continuous variables, formulated as follows:
Maximise:

$$Q = \sum_m \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right] \tag{4.6}$$

subject to: Constraints (4.3-4.5)

$$L_m,\, D_m \geq 0 \quad \forall m \tag{4.7}$$

$$Y_{nm} \in \{0, 1\} \quad \forall n, m \tag{4.8}$$

As input, WeiMod takes an undirected, weighted, or unweighted network and an upper limit for the number of modules and outputs a disjoint partition of the network into several modules. As with MINLP_Mod, WeiMod is non-convex and therefore does not guarantee global optimality. As such the MINLP model is solved iteratively with a different random initial solution for a given number of times and the final partition is that which corresponds to the largest value of modularity. In this chapter it is proposed that the MINLP formulation of modularity optimisation be applied without the improvement stage featured in iMod. Consequently, the MINLP model is now solved 1000 times, as

opposed to 100 times in MINLP_Mod, in order to give a better representation of solution space.

4.4 Results

In this section the performance of WeiMod is compared with three other modularity optimisation methods from the literature on a series of synthetic and real networks. Furthermore, WeiMod is compared with the two-stage clustering approach iMod, in order to determine if increasing the number of times the MINLP is solved compensates for leaving out the improvement stage of iMod.

All implementations of WeiMod were performed using GAMS (General Algebraic Modelling System) [172]. The MINLP is solved using the SBB (standard branch and bound method) mixed integer optimisation solver [1] (see Section 4.4.2.3 for a description) and CONOPT as the default non linear programming (NLP) solver [4], while relative and absolute gaps are set to zero. The algorithm has a computational limit of 100000 seconds where necessary. All experiments were run remotely on a bioinformatics Sun Fire X4450 Server running 16 Xeon(R) E7340 processors at 2.4GHz and 32GB of PC2-5300 667 MHz ECC fully buffered DDR2 memory. The server runs CentOS Linux release 5.8 OS.

4.4.1 Synthetic networks

The efficiency of WeiMod is first illustrated through benchmarking on artificial weighted networks with known community structure, in a similar analysis to that seen in Chapter 3, Section 3.3.1. The synthetic networks are again generated following the design of Newman and Girvan [148], however in this chapter, the networks are weighted. The software used to generate the networks is described in [107]. The networks comprise 128 nodes and are partitioned into four communities of 32 nodes each. As in Chapter 3, the cases of degree equal to 16 and degree equal to 5 are both tested. The performance of WeiMod in revealing the known community structure is compared with CNM [49], Louvain [34] and QCUT [176], three of the most popular modularity maximisation methods. For details of each of these methods, see Chapter 2. For all methods, implementations were downloaded from the corresponding sites [48, 81, 175] and clustering experiments run remotely on a bioinformatics Sun Fire X4450 Server described above.

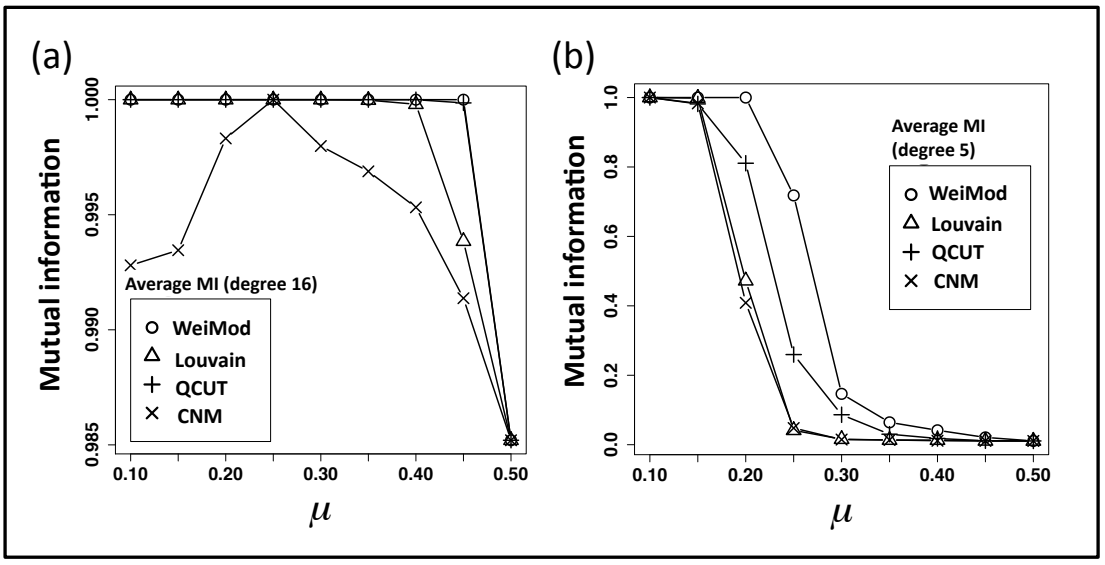

FIGURE 4.2: Benchmarking of module detection performance with WeiMod, CNM, Louvain and QCUT. Synthetic weighted network examples (128 nodes, 4 modules) were generated with node degrees of 16 and 5 in (a) and (b) respectively.

The mutual information measure, described in [107], was used to quantify the agreement between the known and detected community structures. The value of mutual information ranges from 0 for dissimilar, to 1 for identical community structures. See Chapter 3, Section 3.3.1 for more details on the MI calculation. 100 synthetic weighted networks for each value of the mixing parameter, μ, from 0.1 to 0.5 (at intervals of 0.05) were generated with software downloaded from [71]. As μ increases, the modules become more difficult to detect and therefore the ability of a method to uncover the correct community structure is put to the test. Each network is partitioned with the four methods, and the average mutual information calculated for each method for each value of μ. Figure 4.2 reports the average mutual information values against the mixing parameter for the synthetic networks to illustrate how close these methods are in revealing the known community structure.

For degree equal to 16, WeiMod, Louvain and QCUT retrieve the exact partition for all values of μ up to 0.35. This trend continues for WeiMod and QCUT up to 0.4, whereas Louvain achieves a value of mutual information less than 1 at 0.4 and the methods performance starts to decline thereafter. For $\mu > 0.4$, WeiMod marginally outperforms QCUT by continuing to extract the exact partition at $\mu = 0.45$, whereas QCUT's performance starts to decline at this point. CNM performs badly at the start

but manages to achieve mutual information equal to 1 at $\mu = 0.25$, however the methods performance then rapidly declines. In the case of degree equal to 5, WeiMod detects the exact community structure for μ equal to 0.1, 0.15 and 0.2, outperforming all other methods, none of which detect the exact partition for any value of the mixing parameter after $\mu = 0.1$. Although WeiMod's performance declines rapidly from $\mu = 0.3$ onwards, the average mutual information for WeiMod continues to be larger than for all other methods. Overall, WeiMod performs better than the other methods tested.

4.4.2 Real Networks

The synthetic networks generated in the previous section can give a good indication of the accuracy of community structure detection methods, however, these are not realistic network examples. Therefore, the performance of WeiMod on several real life networks is now investigated, again in comparison to CNM, Louvain and QCUT and additionally iMod. Furthermore, the significance of the modularity values detected is verified and finally, the choice of commercial solver is explored.

4.4.2.1 Method comparison

WeiMod is once again compared with CNM, Louvain, and QCUT on a selection of weighted and unweighted biological and social networks. Although the main aim of this chapter is to develop a method to accurately cluster weighted networks, unweighted networks are also included so that comparisons can be made with iMod without having to extend OptMod to weighted networks. Table 4.1 shows a summary of the networks, in terms of number of nodes, number of edges, whether the network is weighted or unweighted and the network density. Density is defined as the actual number of edges divided by the possible number of edges, $D = \frac{2L}{N(N-1)}$, where L is the total number of edges and N is the total number of nodes in the network. The networks tested are briefly described below. Weighted networks tested are:

- Victor Hugos Les Miserables dataset, compiled by Knuth [101], describing interactions between characters in the novel of the same name. Weights represent the number of co-appearances of characters in the novel. The unweighted version was used in the method comparison in Chapter 3 and is also used below to test WeiMod.

- USAir97 describes the direct flight connections between US airports in 1997 [3]. The unweighted version was used in Chapter 3.
- A second transport network, Airports, from the 500 airports with the highest traffic obtained from census data [50]. Weights reflect the actual traffic of passengers on each connection between airports.
- The main component of the weighted protein interaction network of the *Caenorhabditis elegans* worm with weights based on phenotypic profile similarity [186].

Unweighted networks are also tested in the method comparison, these are:

- The popular Zachary karate network describing the relationships between members of a karate club that split into two clubs due to a dispute between the administrator and the trainer [224].
- Lusseau's dolphin dataset describing communications between dolphins during a field study in Doubtful Sound New Zealand [130, 131].
- The unweighted version the Les Miserables character network [101].
- Krebs' political book dataset modelling the co-purchasing of books about US politics from Amazon (Polbooks) [102].
- A network modelling the schedule of football games between teams in an American university league [74].
- The p53 protein protein interaction (PPI) network [53].
- The network of the early secretory pathway (ESP) of budding yeast [181].
- The university email communication network [84].

For each community detection experiment, WeiMod is run 10 times, resetting the seed for the random number generator each time and each of the 10 runs the MINLP is solved 1000 times. Table 4.2 shows the best and median modularity and the average CPU across the 10 runs, as well as the number of modules in the partition corresponding to the best value of modularity. The results from CNM, Louvain and QCUT are also reported in Table 4.2.

Network	**Nodes**	**Links**	**Weighted**	**Density**
Les Miserables (1)	77	254	Yes	0.0868
USAir97	332	2126	Yes	0.0387
Airports	500	2980	Yes	0.0239
***C. elegans* main**	889	22652	Yes	0.0574
Karate	34	78	No	0.1390
Dolphin	62	159	No	0.0841
Les Miserables (2)	77	254	No	0.0868
p53	104	226	No	0.0422
Polbooks	105	441	No	0.0808
American football	115	613	No	0.0935
Jazz	198	2742	No	0.1406
Yeast ESP	400	3416	No	0.0428
Email	1133	5451	No	0.0085

TABLE 4.1: Summary of the networks involved in the method comparison.

	WeiMod				**CNM**		**Louvain**		**QCUT**	
Network	**Best Q**	**Median Q**	**M**	**CPU**	**Best Q**	**M**	**Best Q**	**M**	**Best Q**	**M**
Les Miserables (1) (w)	**0.5667**	0.5472	5	12.47	0.5472	5	0.5654	6	**0.5667**	6
USAir97 (w)	**0.2150**	0.1936	4	303.83	0.1936	4	0.1957	5	0.2100	4
Airports (w)	**0.2855**	0.283	13	1540.78	0.283	13	0.2823	13	0.2847	11
***C. elegans* main (w)**	**0.3737**	0.2924	9	8660.12	0.2924	9	0.3714	12	0.3618	7
Karate	**0.4198**	0.3807	3	1.30	0.3807	3	0.4188	4	**0.4198**	4
Dolphin	**0.5285**	0.4955	4	3.91	0.4955	4	0.5185	5	0.5175	5
Les Miserables (2)	**0.5600**	0.5006	5	12.34	0.5006	5	0.5556	6	**0.5600**	6
p53	**0.5344**	0.5205	8	21.21	0.5205	8	0.5274	7	0.5219	8
Polbooks	**0.5272**	0.502	4	19.49	0.502	4	0.5205	4	0.5208	4
American football	**0.6046**	0.5773	7	43.81	0.5773	7	**0.6046**	10	**0.6046**	10
Jazz	**0.4451**	0.4389	4	127.24	0.4389	4	0.4431	4	0.4445	3
Yeast ESP	**0.2482**	0.2289	7	1277.14	0.2289	7	0.2406	7	0.2405	6
Email	0.5678	0.5633	10	2676.31	0.5148	12	0.5426	11	**0.5762**	12

TABLE 4.2: Comparison of method performance for networks studied. Weighted networks are denoted with (w), all other networks are unweighted. The best modularity achieved over all four methods appears in bold text. The number of modules detected is denoted by M.

Results show that for all but one of the networks (the email network), WeiMod achieves modularity values as good as, but in most cases better, than CNM, Louvain and QCUT. Moreover, WeiMod finds globally optimal solutions for the karate, dolphin, Les Miserables, polbooks and American football networks. QCUT finds globally optimal solutions for the karate, Les Miserables and American football networks and Louvain for the American football network. The global optimum values for these networks can be found in Table 3.2 in Chapter 3. Note that the version of the USAir97 network in Table 3.2 is unweighted and therefore the value of Q reported to correspond to the globally optimal partition is not relevant for the version of the network used in this chapter. In the case of the email network, QCUT detects the partition with the highest value of modularity and WeiMod the second highest. The email network is the largest of all the networks tested and therefore this example illustrates a scalability limitation of WeiMod. It is noted however that WeiMod detects a partition with a higher value of modularity than the Louvain method, a method known for its efficiency in clustering extremely large networks [72] and therefore this is still a relatively positive result for WeiMod. Overall it can be said that WeiMod performs competitively on small to medium sized networks, (the email network at 1133 nodes is classed as a large network).

It is noted here that, as was mentioned briefly in Section 2.3.7, the degeneracy of the solutions seen in Table 4.2 is not surprising as it is a known that the structure of partitions with modularity values close to that of the optimal partition can vary significantly [76]. This is discussed further in Section 8.4.

Overall, discounting the email network, WeiMod performs as good as or better than the other methods for networks up to size 889 nodes. Furthermore, WeiMod finds the highest value of modularity for all of the weighted networks, achieving one of the main aims of this chapter.

WeiMod is now compared with MINLP_Mod and the two-stage iMod algorithm. In a complete run of iMod, the MINLP model in Stage 1 is solved 100 times to detect a good initial partition of a network before being improved in the second stage. Here however, the MINLP model in WeiMod is solved 1000 times in order to try and compensate for the fact that the improvement stage is not included. The results for MINLP_Mod and iMod discussed here can be found in Tables 3.3 and 3.6 in Chapter 3. WeiMod, MINLP_Mod and iMod were all applied to the karate, dolphin, Les Miserables, p53, polbooks, American football and email networks.

Globally optimal solutions are known for all of the above networks except for the email network (Table 3.2 in Chapter 3). These networks are discussed first. WeiMod uncovers a sub-optimal partition for the p53 network, but globally optimal solutions for the remaining five networks. MINLP_Mod does not detect globally optimal solutions for the p53 and American football networks, but does for the remaining four networks. And as has been shown in Chapter 3, iMod achieves globally optimal solutions for all six of the networks. Therefore, in the case of the American football networks, solving the MINLP a larger number of times allows the optimal solution to be found without the help of the improvement stage. However in the case of the p53 network, although WeiMod finds a larger value of modularity than MINLP_Mod it still does not achieve global optimality, illustrating the value that Stage 2 in iMod contributes to the solution procedure.

In the case of the email network, WeiMod has already been shown to perform less well than the QCUT method. Here, it is found that WeiMod achieves a higher value of modularity than MINLP_Mod, which is to be expected due to the larger number of times the MINLP is solved. However, iMod outperforms both WeiMod and QCUT, again indicating that increasing the number of times the MINLP is solved does not sufficiently compensate for the improvement stage in iMod, suggesting that WeiMod could benefit from such an improvement step.

The performance of WeiMod is now compared with MINLP_Mod and iMod in terms of computational cost. The average CPU values across 10 runs for the three methods are shown in Table 4.3 and the computational cost of WeiMod is seen to more or less increase exponentially with network size. Clearly WeiMod has a higher computational cost than MINLP_Mod as the optimisation problem is solved 1000 times instead of 100 times. For the first four networks in Table 4.3, iMod is faster than WeiMod. However, for the polbooks and American football networks, iMod is approximately 15 and 30 times slower than WeiMod respectively, with both methods detecting globally optimal solutions for these networks. Less extreme is the case of the email network, where iMod achieves a higher value of modularity in just over double the time it takes for WeiMod to terminate. More specifically, the American football network is a relatively small network with only 115 nodes. WeiMod finds the globally optimum partition in 43.81 CPU seconds compared with 1332.86 seconds for iMod. As has been discussed in Chapter 3, the CPU time for iMod is not correlated solely with the number of nodes and edges in a network due to the nature of the iterative improvement stage. Consequently, CPU time can

Network	**WeiMod**	**MINLP_Mod**	**iMod**
Karate	1.30	0.28	0.35
Dolphin	3.91	0.90	2.52
Les Miserables	12.34	2.42	4.75
p53	21.21	2.12	4.46
Polbooks	19.49	4.33	310.75
American football	43.81	19.77	1332.86
Email	2676.31	330.99	4111.62

TABLE 4.3: Comparison of CPU for WeiMod, MINLP_Mod and iMod on 6 unweighted networks.

sometimes be highly disproportionate to the size of the network, as demonstrated by the American football network example above.

Therefore again, the results point to the dilemma of accuracy versus efficiency and the question of which can be sacrificed. As in Chapter 3, the general aim in this chapter is to develop methods that achieve high values of modularity. Therefore it is concluded that the MINLP formulation alone is insufficient and Stage 2 of iMod should also be generalised to weighted networks in order to improve solutions. However, the issue of the instability of the computational cost incurred by Stage 2 of iMod should not be ignored and as such perhaps future work should also focus on ways of improving the efficiency of WeiMod, rendering it more applicable to larger networks.

Finally, to illustrate the effect that the inclusion of weighted interactions has on solutions found by modularity optimisation, Figure 4.3 shows the partitions of the weighted and unweighted Les Miserables networks as detected by WeiMod. The difference in node-module allocation underlines the influence that the edge weights can have on the community structure, in agreement with [65], confirming the importance of including weighted edges wherever possible in order to achieve more accurate solutions. Moreover, it is noted in Table 4.2 from the example of the weighted and unweighted Les Miserables networks that including weighted interactions does not affect CPU time.

4.4.2.2 Randomisation of real networks

The modularity metric is based on the assumption that a random network does not exhibit community structure. Modularity is therefore calculated by comparing the density of intra modular edges in the network in question with the expected number of intra modular edges in the corresponding random network. That is, a network with the same

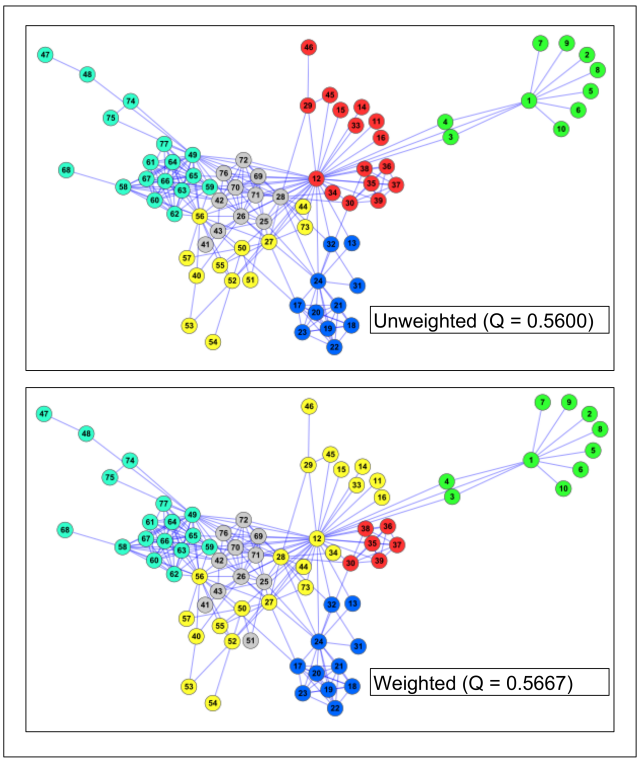

FIGURE 4.3: The partitions of the weighted and unweighted versions of the Les Miserables network, as detected by WeiMod, illustrating that including weights effects the community structure.

number of nodes and the same node degree distribution, but in which the edges have been randomly re-wired. However, it has been shown that some random networks can exhibit community structure by chance [85, 169]. Consequently, it is more accurate to say that a network has community structure if and only if it has a partition with a larger modularity than that of its randomised counterpart. Here, in order to determine the significance of the modularity values found by WeiMod in the previous section, the observed value of modularity is compared to the distribution of modularity values from corresponding random networks with the same size and same degree distribution.

For each network in Table 4.1, 500 random networks were generated using the link reshuffling function, "rg_reshuffling_w", from the tnet package in R [156]. This is carried out by randomly selecting two edges of a network and rewiring them by swapping interaction partners thus creating two new edges, with weights remaining attached to the reshuffled edges. If either of the new edges already exists, the original edges are reformed and two new edges are selected. The procedure is repeated a number of times until a

Network	**Average Q of random networks**	**St. Dev. of random networks**	**Observed Q (WeiMod)**	**z-score**
Les Miserables (1)	0.4261	1.56E-02	0.5667	9.01
USAIR97	0.2656	4.45E-03	0.2150	-11.35
Airports	0.3688	6.29E-03	0.2855	-13.24
C.elegans	0.1045	8.77E-04	0.3737	306.86
Karate	0.3044	1.47E-02	0.4198	7.87
Dolphin	0.3709	1.12E-02	0.5285	14.11
Les Miserables (2)	0.2896	8.02E-03	0.5600	33.73
p53	0.4041	8.71E-03	0.5344	14.97
Polbooks	0.2890	6.64E-03	0.5272	35.90
Football	0.2757	5.46E-03	0.6046	60.22
Jazz	0.1400	2.06E-03	0.4451	148.26
Yeast ESP	0.1727	3.99E-03	0.2482	18.93

TABLE 4.4: Partitioning results of the test networks and their corresponding randomisations.

new random network has been generated. Each random network was partitioned using WeiMod. In order to increase the speed of computation, the MINLP was only solved 100 times (instead of 1000), thereby giving an estimate of modularity. The z-score is used to measure the deviation of the modularity of the observed network from its random expectation [208]. A z-score of greater than 1.96 or less than -1.96 is associated with a p-value of 0.05. Results in Table 4.4 show that in all but two cases (USAir97 and airports), all of the networks tested do indeed exhibit true community structure, with z-scores greater than $+1.96$.

Commonly a value of 0.3 is used to indicate the presence of community structure [49], however it is only truly present if the value of modularity is larger than the corresponding randomly re-wired network. In the case of USAir 97 and Airports, using this cut off would give a true estimate of the significance of the partition found, i.e. these networks do not have a significantly modular topology. However, for the Yeast ESP network, despite a partition with a value of modularity lower than 0.3, the network still proves to be significantly modular. Therefore the presence of community structure cannot be determined simply by enforcing the above threshold.

4.4.2.3 Solver comparison

There are two approaches to solving MINLP models. First, the Branch and Bound (BB) method can be employed, where the relaxed MINLP (RMINLP) is solved. That is, the integrality constraint is relaxed and the problem is solved as a non-linear programming (NLP) model. If one of the Y_{nm} variables is non-integer, then the mathematical model is split into two sub-models by restricting the non-integer Y_{nm}. The sub-models are solved using the local NLP solutions as the bounds and the bounds on the discrete variables are continually adjusted. Each time the bounds on the region are tightened a new NLP is solved starting from the optimal solution of the previous NLP. The procedure stops when the current feasible region is reduced to a single element, or when the upper bound for the region matches the lower bound.

Alternatively, Outer Approximation (OA) can be used, where the MINLP model is linearised and the resulting mixed integer programming (MIP) model is solved which gives a suggestion for the discrete variables, Y_{nm}, (which in the case of WeiMod is a binary variable but may equally be specified over a countable set). Then an NLP model is solved for the fixed Y_{nm} found by the MIP. A further linearisation is then created using the new solution from the NLP. The procedure stops when the MIP is infeasible or the NLP solution deteriorates. In summary OA relies on linearisations to reduce each sub-problem to a smaller feasible set.

OA works well when the MINLP model has a significant combinatorial component and the NLP models are fairly easy to solve. The BB approach works well on models with few discrete variables but significant nonlinear components. All MINLP and NLP in principal are global optimisation problems. Both OA and BB guarantee global optimality under generalised convexity.

The results in Sections 4.4.1 and 4.4.2.1 were generated using the SBB MINLP solver, which implements the BB solution procedure [1]. SBB is based on a combination of the standard branch and bound method and a standard NLP solver (e.g. CONOPT, MINOS or SNOPT). For WeiMod, SBB was used with CONOPT, which is the default NLP solver for SBB in GAMS. A random initial partition is provided and the RMINLP is solved. If the RMINLP model is unbounded or infeasible, or if it fails, SBB will stop. If all the Y_{nm} happen to be integers then this will be returned as the optimal integer solution. Otherwise, then the solution found from solving the initial RMINLP is stored and the Branch and Bound procedure begins, where bounds on discrete variables are

tightened to new integer values based on the current non-integer solutions. The WeiMod model is non-convex and therefore SBB produces a local optimum. As such, the MINLP is solved a number of times to get a good representation of solution space and the largest value of modularity is chosen as the final solution.

In GAMS, outer approximation can be implemented through the DICOPT solver [5], with default MIP solver as CPLEX (see Section 2.3.6 for a description) and default NLP solver as CONOPT. According to the outer approximation method, the NLP (i.e. binary variables become continuous) is solved and the MINLP model is then linearised around this point and then the linear MIP is solved. The discrete variables are then fixed at the optimal values from the MIP model and the resulting NLP is solved. The process is repeated until the MIP is infeasible or the NLP solution deteriorates.

To determine whether OA would lead to WeiMod achieving better solutions, each network in Table 4.1 is partitioned again by WeiMod but with DICOPT as the MINLP solver. Again each experiment was run 10 times, resetting the seed for the random number generator for each run. A comparison of the new results is made with those generated using the SBB solver. Table 4.5 shows the best and median modularity and the average CPU over the 10 runs. The results are analysed in terms of modularity value detected and in terms of CPU. Results show that overall SBB performs marginally better than DICOPT. SBB finds as good as or better modularity for all networks except the Yeast ESP network and SBB is faster for all networks except the C. elegans, Yeast ESP and email networks. This is in agreement with a study showing SBB to perform slightly better then DICOPT [111] when tested on the 250 models in the MINLP world problem library.

For the moment, SBB appears to produce sufficiently good results with WeiMod proving itself to be a competitive method against other modularity optimisation methods (Table 4.2). SBB and DICOPT solvers have been tested in this chapter, however many other MINLP solvers exist, for example, Alpha ECP and BARON (see [6] for a comprehensive list). Furthermore, it has been found that combining solvers can help to improve method performance [111]. Consequently, future work will involve the investigation of alternative MINLP solvers and combinations of such solvers to determine whether WeiMod's efficiency can be improved in this manner.

Network	SBB			DICOPT		
	Best Q	**Median Q**	**CPU**	**Best Q**	**Median Q**	**CPU**
Les Miserables (1)	0.5667	0.5667	12.47	0.5667	0.5667	17.97
USAir97	0.2150	0.2130	303.83	0.2144	0.2136	338.62
Airports	0.2855	0.2855	1540.78	0.2855	0.2854	1837.05
***C. elegans* main**	0.3737	0.3726	8660.12	0.3730	0.3725	6223.58
Karate	0.4198	0.4198	1.30	0.4198	0.4198	5.07
Dolphin	0.5285	0.5285	3.91	0.5285	0.5285	10.33
Les Miserables (2)	0.5600	0.5600	12.34	0.5600	0.5600	28.86
p53	0.5344	0.5296	21.21	0.5333	0.5292	49.41
Polbooks	0.5272	0.5272	19.49	0.5272	0.5272	37.26
American football	0.6046	0.6046	43.81	0.6046	0.6046	52.59
Jazz	0.4451	0.4451	127.24	0.4451	0.4451	177.97
Yeast ESP	0.2482	0.2476	1277.136	0.2492	0.2476	419.326
Email	0.5678	0.5633	2676.31	0.5636	0.5608	2393.75

TABLE 4.5: Comparison of results found by WeiMod when using the SBB solver and the DICOPT solver.

4.5 Alternative objective functions

One of the reasons that mathematical programming has been employed as the modelling framework in this thesis is the flexibility it offers. It has been seen in this chapter how incorporating weighted interactions into an existing MINLP model required a relatively simple adjustment. It is now proposed that WeiMod can act as a template model for further clustering methods that tackle various aspects of the community structure detection problem. In this section, objective functions are described that (i) address the clustering of directed networks and (ii) offer potential solutions to the resolution limit problem exhibited by modularity optimisation (mentioned in Chapter 2, Section 2.3.7 and will be discussed in more detail in Chapter 8). Further investigation into new clustering methods that incorporate these objective functions will feature in future work.

4.5.1 The modularity metric for directed networks

Modularity has a straight forward extension to directed networks where one no longer considers simply the degree of a node, but its in- and out-degree [117]. For node n, the in-degree is the number of connections that other nodes in the network make with n that are directed towards node n and the out-degree is the number of connections node n makes that are directed towards other nodes. Modularity for directed networks

is defined as follows:

$$Q = \frac{1}{L} \sum_{ij} \left[A_{ij} - \frac{d_i^{in} d_j^{out}}{L} \right] \delta(c_i, c_j) \tag{4.9}$$

where, m is the total number of edges in the network, A_{ij} is the weight between nodes i and j, d_i^{in} is the in-degree for node i and d_j^{out} is the out-degree for node j. Finally, $\delta(c_i, c_j)$ is the Kronecker delta symbol, which is equal to 1 if nodes i and j are in the same module, 0 otherwise. For weighted networks, in- and out-degree can be replaced by in-strength and out-strength as defined previously for weighted node degree. The generalisation of WeiMod to directed networks would accommodate applications to biological networks with directed interactions. For example, in a transcription network, nodes represent genes, and a directed edge from gene X to gene Y indicates that the transcription factor encoded by gene X regulates the transcription of gene Y. An example of such a network is the transcriptional regulation network of *Escherichia coli*, described in [183].

4.5.2 Solutions to the resolution limit problem

It has been noted that modularity optimisation suffers from a resolution limit [73]. This involves combining smaller communities into larger ones to achieve better modularity and therefore modules smaller than a certain scale may not be found. For more details, see Chapter 8, Section 8.4. Here, two potential solutions that can be easily incorporated into the existing WeiMod mathematical model are presented. First, in [121] the quantitative function, Modularity Density, D, is introduced which is related to the average degree of the nodes inside a module and the fraction of edges that leave the module. Modularity density is defined as follows:

$$D = \sum_{m} \frac{L(V_m, V_m) - L(V_m - \overline{V_m})}{|V_m|} \tag{4.10}$$

where $L(V_m, V_m)$ is the number of edges that lie fully within module m, $L(V_m - \overline{V_m})$ is the number of edges formed between nodes in module m and nodes outside of module m and $|V_m|$ is the total number of nodes in module m. For weighted networks, the number of edges in this calculation can be replaced by the sum of the weights of the links that lie within module m etc. The authors of [121] propose optimisation of modularity density as a viable alternative to traditional modularity for partitioning networks as it does not encounter the same resolution limitations.

Alternatively, in [119], it is suggested that one can probe the community structure of a network at multiple scales. Modularity is adapted to include a resolution parameter, λ, to determine the scale of community structure detected by the clustering algorithm as follows:

$$Q_{res} = \frac{1}{2L} \sum_{ij} \left[A_{ij} - \lambda \frac{d_i d_j}{2L} \right] \delta(c_i, c_j) \tag{4.11}$$

where L is the total number of edges in the network, A_{ij} is the weight between nodes i and j, d_i is the degree/strength of node i and similarly for d_j and $\delta(c_i, c_j)$ is the Kronecker delta symbol, which is equal to 1 if nodes i and j are in the same module; 0 otherwise. For $\lambda = 0$, all nodes belong to the same module and for $\lambda = 1$, Q_{res} is the same as the original Newman modularity[148]. As λ increases, communities split and become smaller, until all nodes belong to individual communities. Through the investigation of multiple scales of community structure in protein interaction networks, the authors of [119] conclude that all scales of community structure may be biologically meaningful. Therefore partitions detected by modularity optimisation should not be discounted despite the known resolution limit problem, as each level of community structure may be informative in its own right.

4.6 Discussion and conclusions

In this chapter, Stage 1 of the iMod clustering procedure from Chapter 3, MINLP_Mod, was generalised to detect disjoint community structure in weighted networks. In addition to weighted interactions, the new methodology, known as WeiMod, also accommodates loops, i.e. self-interactions. These changes allow for a more realistic abstraction of the system under study. For example, in biological networks, weights may represent the strength of correlation between gene expression profiles and loops can model auto-regulation of a transcription factor. Including such information, or alternatively choosing not to include it, will generally impact on the resulting community structure and therefore the accuracy of solutions.

It was also proposed that the MINLP formulation of modularity optimisation could act as a stand-alone clustering method without the need to include the improvement stage featured in iMod. In order to improve on solutions found by MINLP_Mod in Chapter 3, the number of times that the MINLP is solved for a single clustering experiment was increased from 100 to 1000. The aims of this chapter were therefore to (i) show

WeiMod to be competitive in comparison with methods from the literature, in particular on weighted networks and (ii) determine whether WeiMod could perform as well as iMod despite the lack of improvement stage.

WeiMod was compared against three well-known modularity optimisation methods on a series of weighted and unweighted networks of varying sizes. WeiMod detected modularity values that were as good as or better than those found by all other methods for networks of up to 889 nodes, achieving globally optimal solutions on several of the benchmark networks. In particular, WeiMod outperformed all other methods on all weighted networks tested, achieving one of the main aims of this method re-formulation. However, WeiMod began to show signs of limitations when clustering the email network, the largest network tested (1133 nodes), indicating the need for improvements to scalability.

A comparison was also made with MINLP_Mod and iMod. It was found that solving the MINLP 1000 times instead of 100 either achieved the same or improved on results found by MINLP_Mod. However, it was also shown that iMod achieved better modularity values on two of the networks. In particular, iMod detected a partition of the email network with a larger modularity than the partitions found by WeiMod and the three modularity optimisation methods from the literature. These results indicate that, based on modularity value alone, iMod performs better than WeiMod. The question is therefore, should Stage 2 of the iMod procedure be extended to weighted networks and be included as an improvement step after the application of WeiMod? If deciding based solely on the values of modularity detected, then the answer is yes. However as discussed in the results section, the disproportionate CPU time sometimes incurred by Stage 2 in iMod should also be considered. It is concluded that there are three avenues to explore in terms of future method development: (i) extending OptMod in the same manner as MINLP_Mod and creating a two-stage algorithm to cluster weighted networks, (ii) investigate means of increasing the efficiency and accuracy of WeiMod alone and (iii) explore parallel computation.

Possible improvements in efficiency may come from the use of alternative solvers. This is partially investigated in this chapter by considering the results generated by SBB with those generated by DICOPT. Although in this case significant differences were not found, future work will feature the investigation of additional solvers and moreover the possibility of combining solvers. Furthermore, improvements in efficiency may be

possible through the implementation of symmetry breaking constraints as seen used in OptMod [219].

Overall the methodology presented in this chapter represents a small step in the direction of developing more realistic and informative modelling frameworks. The search for more accurate models does not stop here; detailed network representations that include directionality, alternative fitness functions, and the incorporation of dynamic features can all contribute to future advances in community structure detection. The development of such models will be facilitated by the flexibility provided by employing a mathematical programming framework. In the following chapter, the next step in the evolution of the methodology presented in this thesis is taken by considering the detection of overlapping communities.

Chapter 5

Detecting overlapping community structure in complex networks

As has been discussed throughout this thesis, community structure detection has proven to be an important analytical tool in various areas of research. However, the standard problem of detecting a partition of disjoint modules has been continually adapted to model more closely the intricate relationships in real life complex systems. For example, this has been done through the development of community structure detection methods applicable to networks with weighted interactions, a problem that has been addressed in Chapter 4. In this chapter, by incorporating the concept of overlapping communities into the modelling framework, the pursuit of more information rich models of complex systems is continued. Here, an MINLP method for the transformation of disjoint communities to overlapping communities is proposed. A method evaluation is carried out on a small benchmark network and a comparative analysis is made with methods from the literature. The performance of the methodology is further assessed on protein-protein interaction (PPI) networks to test the method's ability to extract meaningful biological results. Results show that proteins assigned to more than one module by the method exhibit properties indicative of their relevant role in the organisation of the entire system.

5.1 Introduction

Just as weighted interactions naturally occur in many complex systems, so does the idea of a node belonging to more than one community. For example, in protein interaction networks, disjoint community structure detection has found functionally coherent modules [119, 201]. However, in reality, some proteins carry out more than one task or belong to more than one protein complex [103], a property which, when accommodated, may lead to more accurate solutions.

Modules are widely regarded as semi-independent functional units within an entire system. Overlapping modules can be thought of as a means of allowing systems to coordinate the different tasks being carried out by each module. Additionally, overlapping modules illustrate the multi-functionality of a node, the idea of a node acting as a bridge between different functional groups and its role in helping to maintain the structural cohesiveness of the system. Nodes with such structurally relevant roles can be seen in social networks modelling the spread of disease where potential immunisation targets are the individuals that bridge communities [179]. Here we ask if these ideas translate to a biological context and do similarly positioned biomolecules also adopt such important strategic roles? And consequently, how can such nodes be detected?

The question of how to find overlapping communities does not have a straightforward answer. The overlapping community structure detection problem can be interpreted differently according to experimental requirements and as a result existing methods vary to a large degree. Equally, due to the lack of a gold standard, evaluating method performance and comparing results across methods becomes a complex process. Due to the introduction of the Newman modularity measure [148], there is to some extent an agreement on the definition of the disjoint community structure detection problem and a benchmark for method development. Methods generally aim to find a partition of a network into densely connected modules, in many cases via the optimisation of modularity. Apart from the criteria that a node belonging to two modules must be connected to nodes in both modules, the problem statement of detecting overlapping communities is much less well defined. The first challenge is therefore to decide how to interpret the problem in the context of this thesis and define a suitable solution procedure.

As mentioned above, in the network modelling the spread of disease, nodes at the intersections of modules are structurally relevant in the global functioning of the system. It is

shown that targeting these nodes can reduce the spread of disease. Translating these concepts to a biological context, the aim becomes finding biomolecules that link functional units and that if removed or targeted in some way would to a degree disrupt the global functioning of the system. Equally, this can be thought of as identifying biomolecules that due to their position in the network contribute towards the maintenance of the status quo. Therefore, the general overlapping community structure detection problem is to detect nodes belonging to more than one module that play an integral part in the overall functioning of the system. More specifically, it is desirable that these nodes display topological and functional properties reflecting their importance as connectors within an entire system.

The above problem can be thought of as the next step in the continually evolving community structure detection problem and as such becomes an extension of previous work. The approach proposed is therefore a two-stage procedure. First the disjoint communities of a network are detected using standard clustering methods, allowing previous work to be capitalised on. In stage two, the nodes that form interactions across community borders are examined to assess their associations with modules other than their own. Depending on their potential contribution to the internal structural cohesion of each module, these nodes are then either assigned to multiple communities or remain a member of a single community.

The remainder of this chapter unfolds as follows. First an overview of existing methods is given to illustrate the variety of approaches that have been employed to tackle the problem. An MINLP model is then proposed to convert disjoint communities of a network to overlapping communities. The methods performance is first assessed on the small benchmark karate network and a comparison is made with results from the literature. The method's ability to extract meaningful biological results is then evaluated by considering the overlapping community structure of the PPI networks of rat and human. Properties of proteins belonging to more than one module are investigated in order to determine whether the proposed method can indeed assign structurally and functionally relevant nodes to multiple modules.

5.2 Background and related work

Before giving an overview of existing methodology, some concepts are introduced to make clear any distinction between overlapping communities and disjoint communities. A node

that belongs to more than one module is said to be multi-clustered, whereas a node belonging to a single module is said to be mono-clustered. A cover is the decomposition of a network into overlapping modules, where nodes can be assigned to multiple modules, as opposed to a partition where all nodes belong to only one module. A partition is also known as a hard partition and a cover is known as a soft partition. In some models, nodes with multiple module membership may participate in each of their modules with a measure of the strength of belonging, known as the belonging coefficient (BC). In this way, a node may belong more strongly to one community than another [153, 205, 226]. These concepts feature in the methods described in this section and throughout the remainder of the chapter.

Although a less well-covered area of research than standard community structure detection, several methods for detecting overlapping communities have been proposed. As has been mentioned previously, the problem is open to interpretation and consequently approaches vary considerably. An overview of some of the existing methods is now given.

One of the pioneering methods for detecting overlapping communities is the Clique Percolation method, known as CFinder, proposed by Palla et al. [159]. The method finds a set of k-cliques where a k-clique-community is defined as the union of all k-cliques that can be reached from each other through a series of adjacent k-cliques. Since vertices can belong to more than one k-clique, overlapping communities are produced. A disadvantage of this method is that it is not clear which value of parameter k (where k defines the number of nodes in the cliques) should be chosen to find optimal solutions. Additionally, in some cases, nodes may not be assigned a community and therefore for some applications may not offer satisfactory or relevant solutions. After the introduction of CFinder, many more methods followed.

In [182], Shen et al. proposed a method based on constructing a maximal clique network (with k-cliques) from the original network where a detected clique becomes a meta-node of the maximal clique network. It is then shown that detecting disjoint communities in the maximal clique network is equivalent to finding an overlapping partition of the original network. k-cliques are further employed in [140] as initial community cores, which are then merged based on the increase in a modified definition of modularity that produces overlapping communities. In [215], Wu et al. use a similar method to Shang et al. based on optimising the modularity of the maximal clique network in order to investigate the overlapping communities of the structural brain network. Clique

detection is also used to detect overlapping communities on the "communicability graph" of the network in [64].

In [211], Wei et al. first find an initial partition using a spectral bisection method with multi-level recursion and the initial seed sets are extended using a lazy random walk. Both Pizzuti [161] and Nicosia et al. [153] use genetic algorithms to detect overlapping modules. Pizzuti runs the algorithm on the line graph of the original network, optimising a defined community score. Nicosia et al. optimise a version of Newman's modularity that has been extended to incorporate overlapping communities. Similarly, Chen et al. [42] optimise an adapted modularity measure, based on choosing an initial seed and expanding based on the strength of the belonging coefficients. In [30] Becker et al. propose Overlapping Community Generator (OCG), a greedy agglomerative method. OCG is based on an adapted modularity measure applicable to overlapping communities. An initial partition of centred cliques is generated, then elements are joined together progressively based on maximising the increase in the average modularity gain. Becker et al. find a cover of the human PPI network with OCG and investigate properties of the multi-clustered proteins.

Newman and Girvan's well-known GN algorithm [74], the original modularity optimisation method (Chapter 2, Section 2.3.1), has been modified by Gregory [79] such that vertices can belong to more than one module by splitting themselves according to the spilt betweenness criteria. Similarly Newman's greedy agglomerative method [146] (Chapter 2, Section 2.3.2) has been extended in [206] where the greedy algorithm detects a disjoint partition and then nodes are assigned more than one module depending on their contribution to the local modularity. The applicability of the method is demonstrated on the yeast PPI network. A similar conversion procedure from disjoint to overlapping communities is described in [216].

In [226], Zhang et al. employ non-negative matrix factorisation (NMF) to detect overlapping communities using a feature matrix generated by normalising the kernel matrix of the network Laplacian. NMF methods are also proposed by Zarei et al. [225] and Lai et al. [106]. Yu et al. [223] propose Approximate Minimum Degree (AMD) Ordering of the adjacency matrix followed by Cholesky factorisation, with a sliding window along the diagonal of the factorised matrix to reveal communities. The method is used to explore the overlapping community structure of yeast and human PPI networks. In [15], communities are defined as groups of links rather than nodes where links are merged

into communities according to the optimisation of a measure known as Partition Density. This objective function is also optimised in [37] to detect link communities via a genetic algorithm. Further methods include, a spectral method [134], a label propagation technique [80] and algorithms based on Markov random walks [136, 95]. Moreover, the well-known Markov Clustering (MCL) algorithm [200] is extended in [185] to detect overlapping communities. The method is applied to three different yeast PPI networks and modules are shown to be more functionally enriched than those found by methods that detect disjoint modules.

Finally, Lancichinetti et al. [110] and Wang et al. [205] locally optimise the following fitness function, named Community Strength in [205]:

$$CS(m) = \frac{\sum_{i \in m} d_i^{in}}{\left(\sum_{i \in m} d_i\right)^r} \tag{5.1}$$

where $\sum_{i \in m} d_i^{in}$ is the total internal degrees of the nodes in module m and $\sum_{i \in m} d_i$ is the total degree of all nodes in module m. In this context, the value of parameter r controls the extent of overlapping to be detected.

In [110] a community is determined through the maximisation of CS starting from a node, i, using a greedy optimisation technique of adding/deleting nodes to find the natural community of node i. In [205], a hard partition is first found and CS is locally optimised for each module again by adding and deleting nodes. Both methods result in overlapping communities.

It is clear that a wide variety of approaches exist and in forthcoming sections it is shown that this variety in methodology is matched by high variability in results, making a comparative analysis between methods relatively difficult. The first task therefore lies in choosing an appropriate methodology that is suitable for applications in bioinformatics. As mentioned in Section 5.1, the aim of this chapter is to develop a method to detect multi-clustered nodes that connect or bridge semi-independent functional units, with properties reflecting their structurally relevant role. Since it has been shown previously that modularity optimisation can find meaningful results in biological networks [115, 119, 201], it is reasonable to adopt the two-stage procedure as seen in [205, 206, 215], where first a hard partition of the network is detected and then intersections between modules identified. This allows a large chunk of the problem to be addressed by existing well-known and well-tested clustering methods. The second stage deals with assessing the borders of the disjoint communities and finding a suitable method for determining

the node-module membership of nodes that form links between modules. This solution procedure is formulated as a mathematical programming method in the following section.

5.3 A mathematical programming model for converting a partition of disjoint communities to a cover of overlapping communities

In previous work, described in Chapters 3 and 4, modularity optimisation has been formulated as mathematical programming models to detect disjoint communities in complex networks. A natural extension to this work would be to incorporate a version of modularity that is applicable to overlapping modules (e.g. [153]) as the objective function in an existing modelling framework. However the disadvantage of this approach is that such an objective function would have a very large search space, rendering the method inefficient. Consequently, the previous mathematical programming methods presented in this thesis are not extended, but instead a novel mixed integer non-linear programming (MINLP) model, known as OverMod, is proposed. OverMod transforms a hard partition of a network into a soft partition by optimising the sum of a measure known as community strength [110, 205] across all modules in the hard partition. The hard partition can be generated by any appropriate disjoint community structure detection method according to the user's preference or the specific network being analysed.

OverMod comprises a mixed integer nonlinear programming (MINLP) model and a series of post-processing steps to determine multi-clustered nodes and their strength of belonging to their respective modules. As input, the method requires a weighted or unweighted, undirected network and a hard partition of the network obtained from any appropriate clustering method. The procedure is outlined in Figure 5.1. The parameters and indices associated with OverMod are the following:

Indices

n, e nodes

m modules

Sets

B_m border nodes for module m

IS_m isolated nodes for module m

BIS_m $B_m \cup S_m$

Parameters

β_{ne} weight of the link between nodes n and e

α_n weight of the edge node n makes with itself i.e. a loop

d_n strength (weighted degree) of node n

L sum of the weights of all edges in the network

r parameter to control the extent of overlapping

K cut off value for the final selection of multi-clustered nodes

BC_{nm} belonging coefficient of node n in module m

BC_{nm}^{norm} normalised belonging coefficient of node n in module m

Continuous variables

L_m sum of weights of all links among nodes within module m

D_m sum of strengths of the nodes in module m

Binary variables

YS_{nm} node membership in the soft partition; equal to 1 if node n is in module m; 0 otherwise.

The sets IS_m and B_m are defined according to each module, m, in the hard partition of the network. IS_m is the set of isolated nodes; nodes which belong to module m and do not interact with nodes outside of module m. B_m is the set of border nodes; nodes in module m that form links with nodes in other modules. Parameter r controls the extent of overlapping where a small r corresponds to a greater extent of overlapping.

The local measure, community strength of module m, $CS(m)$, has been employed in the detection of overlapping communities in [110] and [205] and is shown in equation 5.1. Here, the measure is defined in terms of the L_m and D_m variables that have been defined in previous chapters:

$$CS(m) = \frac{2L_m}{(D_m)^r} \tag{5.2}$$

where D_m is the sum of the strength (weighted degree) of all nodes in module m and L_m is the sum of the weights of the links that lie fully in module m. The idea is that since isolated nodes do not connect with nodes in other modules, they would make little or no contribution to the community strength of modules other than their own. Consequently, their module membership remains fixed and only border nodes have the possibility of belonging to multiple modules in the course of the conversion procedure from hard partition to soft partition. In other words, for all $n \in IS_m$, YS_m is fixed to 1, and for all $n \in N_m$, YS_m is assigned a random initial value of 0 or 1, therefore reducing the number of variables in the optimisation problem thus reducing computational cost.

Unlike in [110] and [205], here $CS(m)$ is simultaneously optimised for all modules in the input hard partition. It follows that the objective function, CS, is the sum of $CS(m)$ over all modules in the hard partition:

$$CS = \sum_m \frac{2L_m}{(D_m)^r} \tag{5.3}$$

where L_m and D_m are defined as follows:

$$L_m = \sum_{n,e \in BIS_m} \beta_{ne} YS_{nm} YS_{em} + \sum_{n \in BIS_m} \alpha_n YS_{nm} \quad \forall m \tag{5.4}$$

and

$$D_m = \sum_{n \in BIS_m} d_n YS_{nm} \quad \forall m \tag{5.5}$$

where d_n is the strength of node n and is defined as $d_n = 2\alpha_n + \sum_e \beta_{ne}$. Note that similar to WeiMod this transformation procedure accommodates self-interactions.

In order to account for the overlapping aspect, the following constraint allows each node to belong to more than one module:

$$\sum_{m:n \in BIS_m} YS_{nm} \geq 1 \quad \forall n \tag{5.6}$$

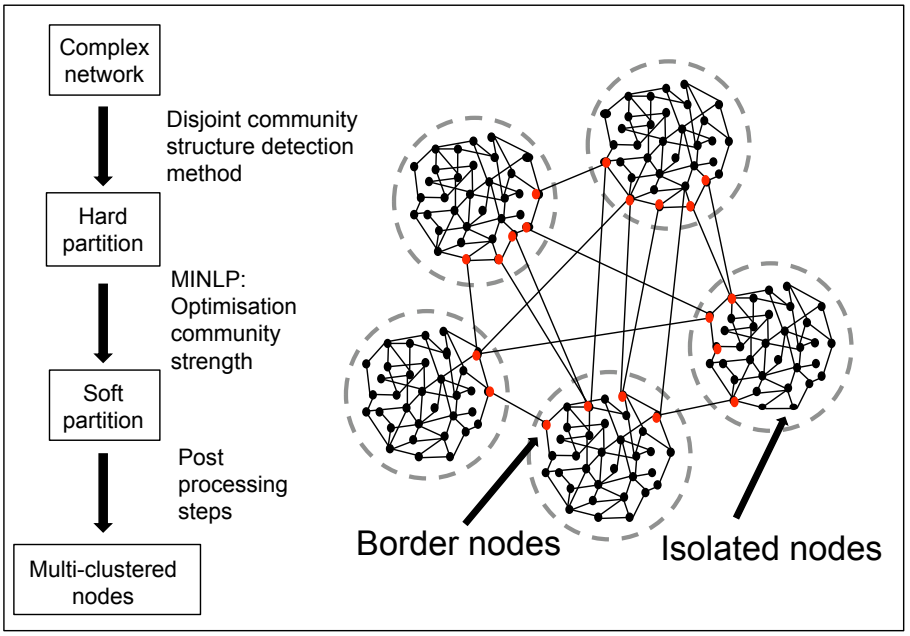

FIGURE 5.1: Outline of OverMod, the conversion procedure from partition of disjoint communities to a cover of overlapping communities. The nodes in red are the border nodes, i.e. nodes that make connections with nodes outside of their own module in the hard partition. These are the nodes that have the possibility to belong to more than one during the conversion procedure. The module memberships of the isolated nodes, i.e. those that only make connections with node in their own community, remain fixed.

The resulting MINLP model comprises a non-linear objective function with a combination of integer and continuous variables, summarised as:
Maximise:

$$CS = \sum_{m} \frac{2L_m}{(D_m)^r} \tag{5.7}$$

Subject to: Constraints (5.4-5.6)

$$L_m,\, D_m \geq 0 \quad \forall m \tag{5.8}$$

$$YS_{nm} \in \{0, 1\} \quad \forall n, m \tag{5.9}$$

Due to the non-convex nature of the model, global optimal solutions cannot be guaranteed. Thus, the MINLP is solved iteratively 100 times, each time with a different random initial solution, giving a good representation of solution space. The largest value of CS corresponds to the best soft partition.

Post-processing steps to obtain the final set of multi-clustered nodes involve calculating the belonging coefficient for each node to each module in the soft partition. This is a measure of strength of belonging of a node to each module. For each module m and each node n in the soft partition, $BC_{nm} = 0$ if either (i) $n \notin B_m$ and $n \notin IS_m$ or (ii) $n \in B_m$ and $YS_{nm} = 0$. Alternatively, if (i) $n \in IS_m$ or (ii) $n \in B_m$ and $YS_m = 1$ for only one module m in the best soft partition, then $BC_{nm} = 1$. Finally, if $n \in B_m$ and $YS_{nm} = 1$ in the best soft partition for more than one module m, the belonging coefficient is defined is the difference between the community strength of the module with node n present and with node n absent:

$$BC_{nm} = CS(m \cup \{n\}) - CS(m \setminus \{n\}) \tag{5.10}$$

where $CS(m)$ is as defined in equation 5.2. Each belonging coefficient is then normalised:

$$BC_{nm}^{norm} = \frac{BC_{nm}}{\sum_m BC_{nm}} \tag{5.11}$$

It follows that any node n with a normalised belonging coefficient not equal to zero or one is a multi-clustered node.

Finally, multi-clustered nodes can then be filtered according to a user-defined threshold, K, where $0 \leq K \leq 1$. The purpose of parameter K is to convert a multi-clustered node to a mono-clustered node when its BC to one of its modules is above the threshold, K, allowing the node to belong fully to its dominant module. This is implemented as follows: for each node n, if the normalised belonging coefficient, BC_{nm}^{norm}, is less than K for all m, then node n is included in the final set of multi-clustered nodes. K equal to 1 is the equivalent of not filtering the results. For example, if a multi-clustered node belongs to two modules, one with BC equal to 0.9 and the other with BC equal to 0.1, it may be desirable to implement the filtering with $K = 0.8$. This would result in the node belonging fully to the first module, which could be seen as the dominant module. Varying the value of K allows the user to determine the level of strength of belonging that would qualify a module to be the dominant module for a multi-clustered node, offering the user more control over the model.

The above procedure can be repeated for a range of values of parameter r. Generally, as r increases, the multi-clustered nodes found for the current r are a subset of the multi-clustered nodes from the previous value of r. However, it is noted that slight discrepancies can appear due to the fact that the MINLP does not guarantee global

optimum. Indicators of an appropriate range of values for r are discussed in forthcoming sections.

All implementations of OverMod were performed using GAMS (General Algebraic Modelling System) [172]. The MINLP is solved using the SBB mixed integer optimisation solver [1] (see Section 4.4.2.3 for a description) and CONOPT as the default NLP solver. The algorithm has a computational limit of 100000 seconds where necessary. As with WeiMod, the relative and absolute gaps are set to zero. All experiments were run remotely on a bioinformatics Sun Fire X4450 Server running 16 Xeon(R) E7340 processors at 2.4GHz and 32GB of PC2-5300 667 MHz ECC fully buffered DDR2 memory. The server runs CentOS Linux release 5.8 OS.

5.4 Computational results on the karate network

In this section some preliminary investigations are made into the performance of OverMod. The problem of detecting overlapping communities is not as well defined as the standard community structure detection problem, for example due to difficulties in conceptualising a uniform definition of overlapping properties. For this reason, methods and parameters used vary greatly and comparisons across different methods and benchmark examples are challenging.

Here, the relatively simple example of the Zachary karate network [224] is used with a view to illustrating similarities and differences between existing methods and OverMod. The two module hard partition, shown in 5.2 (a), was generated with WeiMod, the method outlined in Chapter 4, by setting the upper bound for the number of modules to two. For each value of parameter r in the range 0.7 to 1.1, OverMod was run on the hard partition 10 times, resetting the seed for the random number generator each time. For each of the 10 seeds, the best value of CS and corresponding soft partition were identical and the variance of CS over the 100 solutions was small.

Table 5.1 shows the results of running WeiMod followed by OverMod on the karate network for $0.7 \leq r \leq 1.1$ and $0.6 \leq K \leq 1$. It is noted that the actual two-module hard partition found by Zachary [224] differs slightly from the two-module hard partition found by WeiMod, as can be seen in 5.2 (b). In general, the hard partition used will affect the results produced by OverMod, however in this case the overlaps generated using both hard partitions are identical. From Table 5.1 it is clear that varying the

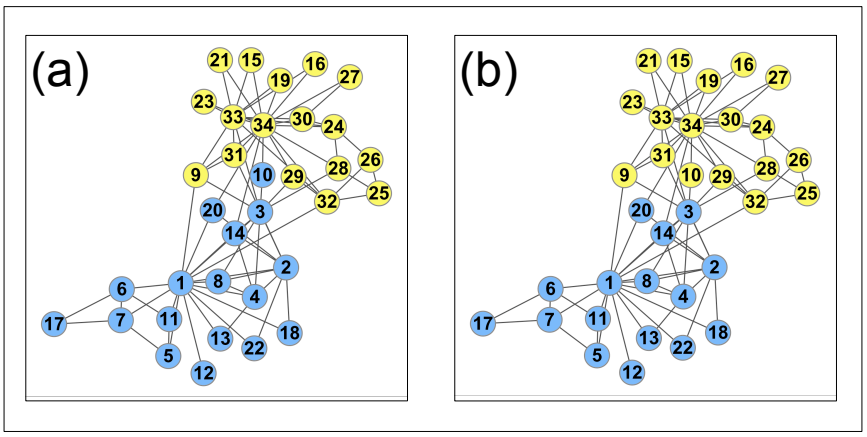

FIGURE 5.2: The two-module hard partitions of the karate network. Partition (a) is the two-module hard partition produced through WeiMod by setting the upper bound for the number of modules to two. Partition (b) is the actual partition of the karate network into two communities as found by Zachary [224].

parameters r and K greatly affects the results. Since the size of the overlap is dependent on the application, it is important to detect covers for a variety of parameter values. The choice of these parameters rests on the requirements of the experiment and should be user guided however, in forthcoming sections criteria that can assist in selecting values of r rationally is discussed.

In order to quantify the variation in the soft partitions found by OverMod and the methods from the literature, the normalised mutual information (MI) proposed by Lancichinetti and Fortunato [107] is employed. MI, taken from information theory, is a measure of similarity between two covers and reflects how much information is needed in order to infer one cover from the other. A value of MI equal to 1 indicates the two covers or partitions are identical. For more details on the MI measure, see Chapter 3, Section 3.3.1. In this case, MI is used to determine which values of the parameters r and K (using all examples in Table 5.1) produce covers with the highest agreement across the methods from the literature. The three cases from the results with the highest average MI across all methods appear on the left hand side column of Figure 5.3. The cover resulting from OverMod with the best agreement across the 8 comparison methods is shown in (a), with the covers with the second and third best agreement in (b) and (c) respectively. The figures to the right of the results found by OverMod are the covers with the same or most similar covers from the literature, according to the value of MI.

r	K	Nodes
0.7	**1**	1, 2, 3, 9, 10, 14, 20, 28, 29, 31, 32, 33, 34
	0.9	2, 3, 9, 10, 14, 20, 28, 29, 31, 32, 33, 34
	0.8	2, 3, 9, 10, 14, 20, 28, 29, 31, 32, 34
	0.7	3, 9, 10, 14, 20, 29, 31, 32
	0.6	3, 9, 10, 14, 20, 29, 31
0.8	**1**	1, 2, 3, 9, 10, 14, 20, 28, 29, 31, 32, 33, 34
	0.9	2, 3, 9, 10, 14, 20, 28, 29, 31, 32, 34
	0.8	2, 3, 9, 10, 14, 20, 29, 31, 32, 34
	0.7	3, 9, 10, 14, 20, 29, 31, 32
	0.6	3, 9, 10, 14, 20, 29, 31
0.9	**1**	2, 3, 9, 10, 14, 20, 28, 29, 31, 32, 33, 34
	0.9	3, 9, 10, 14, 20, 28, 29, 31, 32, 34
	0.8	3, 9, 10, 14, 20, 29, 31, 32
	0.7	3, 9, 10, 20, 29, 31, 32
	0.6	9, 10, 29, 31
1	**1**	3, 9, 10, 31
	0.9	3, 9, 10, 31
	0.8	9
	0.7	9
	0.6	Empty
1.1	**1**	3, 10
	0.9	Empty
	0.8	Empty
	0.7	Empty
	0.6	Empty

TABLE 5.1: Results of the OverMod algorithm on the Zachary karate network for $0.7 \leq r \leq 1.1$ and $0.6 \leq K \leq 1$ with the two-module hard partition in Figure 5.2 (a).

By testing OverMod for a range of parameter values for r and K, there is clearly some strong agreement across the published methods, which can be seen in Figure 5.3. In particular, nodes 3, 9, 10 and 31 appear to be overlapping most consistently, however many additional nodes are also multi-clustered by several methods, illustrating that the variation in methodology can affect results considerably. In the following section a more thorough investigation of the performance of OverMod is carried out and the interpretation of the overlapping community structure problem in the context of biological networks is discussed in more depth.

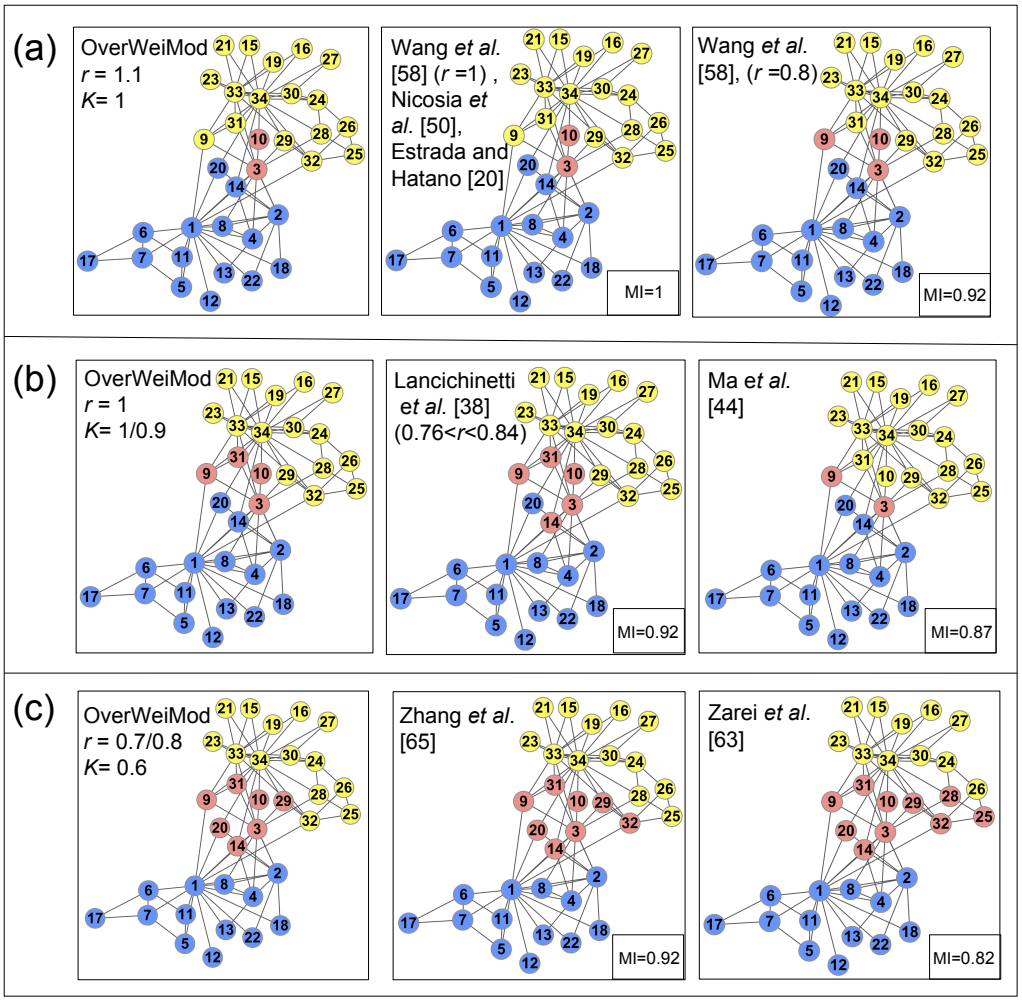

FIGURE 5.3: Covers detected by various methods for the two-module hard partition of the karate network compared with selected OverMod results. Yellow and blue nodes denote the original modules of the hard partition. Nodes with multiple community memberships are shown in red. (a) shows the cover from Table 5.1 that had the highest agreement across the methods, with the closest covers from the literature to the right of it. Similarly (b) and (c) show the partitions with the second and third highest agreement across the methods, respectively, with their corresponding covers from the literature to the right of the figures. The mutual information (MI) values indicate similarities of each cover from the literature with the OverMod cover to its left.

5.5 Exploration of the overlapping community structure of PPI networks

Several methods that detect overlapping community structure have been applied to PPI networks previously [30, 185, 206, 223]. Except from Becker et al. [30], the properties of the individual multi-clustered proteins have generally not been explored, with focus remaining on functional enrichment of modules. In this section, the overlapping community structures of the rat and human PPI networks are investigated. Hard partitions of the networks are detected which are subsequently transformed to overlapping communities via the transformation procedure described in Section 5.3. A closer look is then taken at the characteristics of the proteins that are assigned more than one module in order to determine whether the proposed methodology detects proteins with properties indicative of structurally and functionally relevant roles that allow them to act as connectors between functional modules.

5.5.1 Detecting disjoint community structure

OverMod takes a hard partition of a network and converts it to a soft partition by examining the nodes that make links across community borders. The hard partition can come from any suitable community structure detection method depending on user preference. Both methods used here employ modularity optimisation.

First the rat PPI network is downloaded from BioGRID [191]. The network has 1148 nodes, 1520 links and comprises 118 connected components, including two singletons and 69 pairs. Only the main component, 811 nodes and 946 interactions, is considered in this analysis, as the smaller components have no scope for further investigation into their community structure. A hard partition of the main component was found by iMod, the modularity optimisation method described in Chapter 3, with 22 modules and modularity equal to 0.8445. Figure 5.4 (a) shows the rat PPI network and the meta-network of the hard partition found by iMod. It is clear that there is an uneven spread of module size, and also that there are few connections between nodes in different modules and therefore few border nodes (78). To evaluate the robustness of the clustering method, the network was also partitioned by two other well-known community detection methods: QCUT [176], a spectral method and Louvain [34], a greedy agglomerative method (details of both methods can be found in Chapter 2, Section 2.3). QCUT

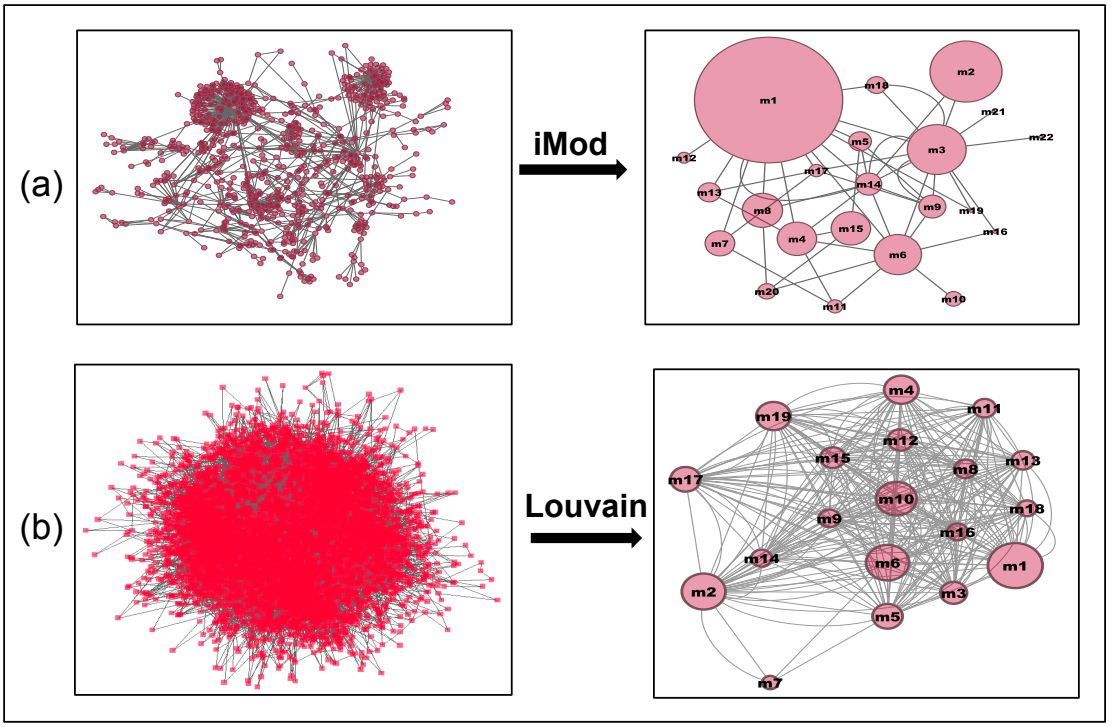

FIGURE 5.4: Detecting the hard partition of the (a) rat and (b) human PPI network with modularity optimisation methods iMod and Louvain respectively.

finds 18 modules with modularity equal to 0.8425 and Louvain detects 19 modules with modularity equal to 0.8429. The module size distributions of the three partitions are shown in Figure 5.5. All three partitions contain modules of less than 100 nodes and one larger module of 169 nodes, supporting the validity of the partition detected by iMod. However iMod slightly outperforms the other two methods in terms of value of modularity and therefore the hard partition found by iMod is used in any further analysis.

The human PPI network, as used in [30], comprises 6171 nodes, 24025 links, 1 main component (6160 nodes and 24014 links) and 3 smaller disjoint components (sizes 3, 4 and 4 nodes). Due to the larger size of the human PPI network, the community structure of the main component is detected by Louvain [34] which is known to be efficient and accurate on very large networks. Louvain finds a partition of the main component into 19 modules with modularity equal to 0.5432. The human PPI network and the meta-network representing the hard partition found by Louvain are shown in Figure 5.4 (b). The human PPI network gives a good example of the hair ball image that one often

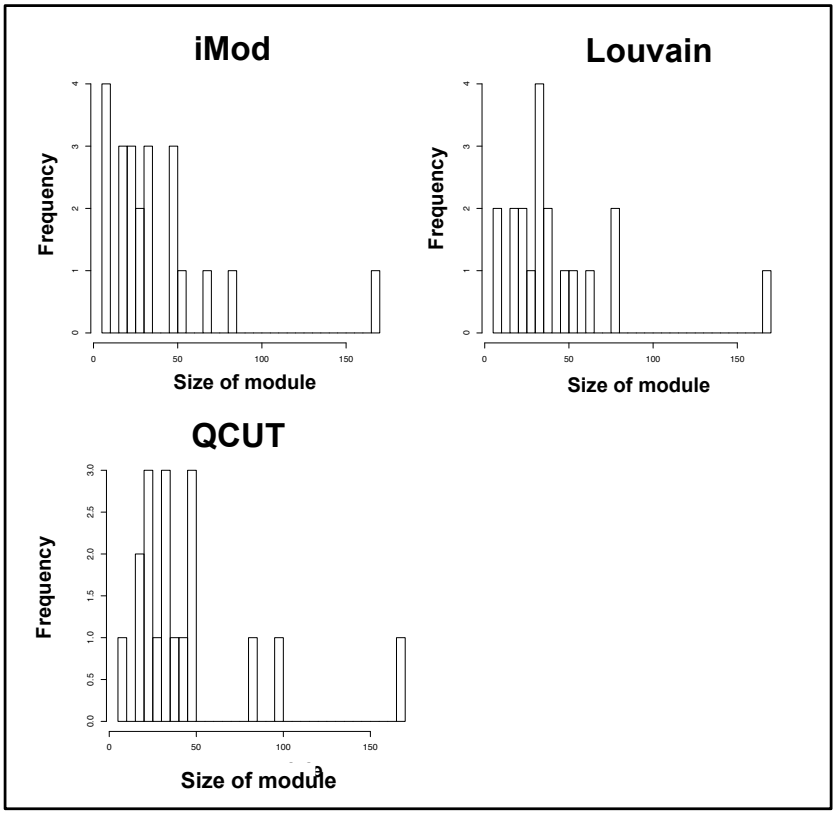

FIGURE 5.5: Module size distributions of the hard partitions of the rat PPI network found by iMod, QCUT and Louvain. All methods detect the module of size 169 nodes as shown on the far right of each histogram.

encounters with visualisations of large complex networks. Such a visualisation does not offer much information regarding the structure of the network, however the meta-network can already offer a greater understanding of the underlying topology. Unlike the rat PPI network, the module size distribution is more even and there are many more links connecting nodes in different modules and therefore many more border nodes (4372). As before, the robustness of the clustering method is evaluated by also partitioning the method with QCUT [176], which detects a partition of 24 modules with modularity equal to 0.5430. Louvain finds all but four modules with < 450 nodes and QCUT finds all but four modules with < 400 nodes, as shown in the module size distributions in Figure 5.6. Based on the fact that Louvain finds a slightly larger value of modularity, this hard partition is chosen for further analysis in the next section.

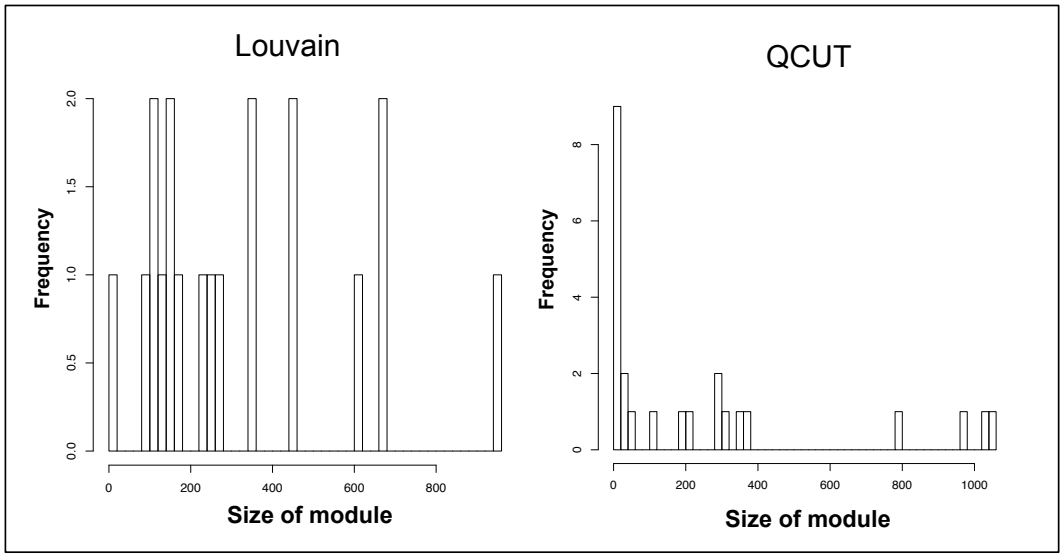

FIGURE 5.6: Module size distributions of the hard partitions of the human PPI network found by Louvain and QCUT.

5.5.2 Converting to overlapping communities with OverMod

First, OverMod converts the hard partition of the Rat PPI network detected by iMod to a soft partition. As mentioned in the previous section, the hard partition results in 78 border nodes (nodes with interactions between modules), which are the potential multi-clustered nodes. That is, the module membership of any non-border node is fixed and the reduced MINLP is solved with only border nodes having the possibility of being assigned to multiple modules. Figure 5.7 (a) shows the number of multi-clustered nodes detected by OverMod for $0.1 \leq r \leq 0.8$ and $0.6 \leq K \leq 1$ in the rat network. The range of r is chosen slightly arbitrarily, however it will be justified in forthcoming sections. Table 5.2 shows the number of modules the multi-clustered nodes belong to for $0.1 \leq r \leq 0.8$. It can be seen that as r increases, not only does the number of multi-clustered nodes decrease, but also the number of modules that they belong to. For $r = 0.1$, multi-clustered nodes are seen to belong to up to 8 modules, however in most cases, the multi-clustered nodes belong to only 2 or 3 modules and for $r \geq 0.6$ multi-clustered nodes belong to at most 2 modules. As K decreases, the number of multi-clustered nodes also decreases, as any nodes with a dominating module are removed from the set.

Similarly, the hard partition of the human PPI network, detected by Louvain, is converted to a soft partition by OverMod. The hard partition results in 4372 border nodes

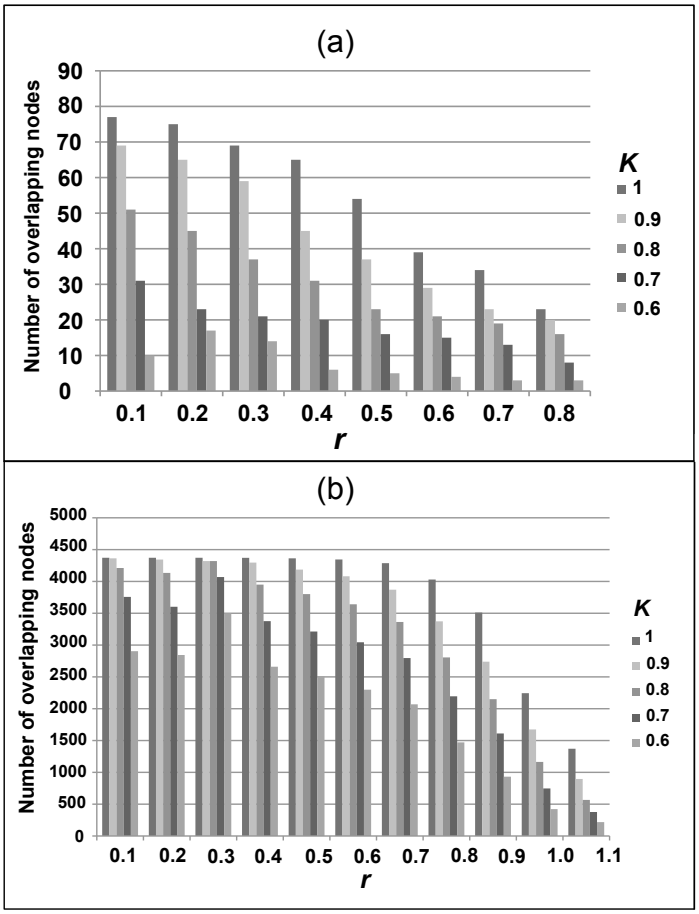

FIGURE 5.7: Multi-clustered proteins in the (a) rat and (b) human PPI networks detected by OverMod for various values of r. For each value of r, multi-clustered proteins are filtered according to threshold K.

	Number of modules multi-clustered nodes belong to						
r	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0.1	60	14	1	0	0	1	1
0.2	59	14	1	1	0	0	0
0.3	57	12	0	0	0	0	0
0.4	59	6	0	0	0	0	0
0.5	50	4	0	0	0	0	0
0.6	39	0	0	0	0	0	0
0.7	34	0	0	0	0	0	0
0.8	23	0	0	0	0	0	0

TABLE 5.2: Distribution of number of modules the multi-clustered nodes detected by OverMod belong to in the rat PPI network.

	Number of modules multi-clustered nodes belong to															
r	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
0.1	1915	1018	524	310	196	132	88	80	35	21	15	12	15	6	3	2
0.2	1915	1018	524	311	195	132	90	80	34	20	15	12	15	6	3	2
0.3	1917	1016	525	311	196	134	91	79	30	21	15	13	14	5	3	2
0.4	1923	1012	526	315	196	137	88	75	27	21	19	11	12	7	2	0
0.5	1940	1022	543	321	202	124	79	55	32	21	14	8	2	0	0	0
0.6	1978	1033	538	305	193	113	74	47	29	19	11	4	1	0	0	0
0.7	2075	1045	505	292	160	93	61	29	22	5	0	0	0	0	0	0
0.8	2243	1016	425	214	90	41	1	0	0	0	0	0	0	0	0	0
0.9	2377	787	246	84	15	2	0	0	0	0	0	0	0	0	0	0
1.0	2019	218	7	0	0	0	0	0	0	0	0	0	0	0	0	0
1.1	1340	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE 5.3: Distribution of number of modules the multi-clustered nodes detected by OverMod belong to in the human PPI network.

and 1788 isolated nodes. Figure 5.7 (b) shows the number of multi-clustered nodes detected by OverMod for $0.1 \leq r \leq 1.1$. The appropriateness of the range will be investigated in forthcoming sections. For $0.1 \leq r \leq 0.9$, more nodes belong to multiple modules than belong to only one module. However, for 1.0 and 1.1, the converse is true, unlike the rat PPI network where the number of multi-clustered nodes is always less than for the mono-clustered nodes. Table 5.3 shows the number of modules the multi-clustered nodes belong to as r changes. For $r = 0.1$ the range is from 2 to 17 modules, and as before, as r increases and the overlap between modules decreases and this range gradually decreases. It is noted that for $0.1 \leq r \leq 0.3$, all border nodes are multi-clustered by OverMod and therefore, for any further analysis only results for $r \geq 0.3$ are considered.

5.5.3 Method comparison with CFinder and OCG

Now a comparison of results is made between OverMod and the overlapping the community structure of the PPI networks found by CFinder [159] and OCG [30].

As already mentioned, the overlapping community structure problem may suffer from multiple interpretations. This is due to the lack of formalisation of the underlying problem statement and additionally because of varying user requirements. A wide variety of methods adopting many different approaches currently exist, as outlined in Section 5.2. This great variation in methodology is reflected in the results, which can be seen even on a small network such as the Zachary karate network, as shown in Section 5.4.

It is concluded that a direct comparison between methods may not be a fair evaluation of performance. Therefore, the aim here is not to compare methods too stringently, but rather to explore the varying results and determine if there is some level of agreement across methods. In any case, for the PPI networks tested (and in fact, most real life networks) the real cover is not known and therefore the aim is to show that according to the user's interpretation of the problem, the chosen method finds biologically relevant solutions. In order to explore the robustness of the approach implemented in OverMod two methods are investigated: CFinder [159] and the Overlapping Cluster Generator (OCG) method [30]. CFinder was chosen due to its status as a method pioneering overlapping community structure detection and OCG as it has been specifically applied to PPI networks and been shown to detect multi-functional proteins. Additionally, each method has an implementation that is readily available from the authors. Some details of the methods are given in Section 5.2.

First, to illustrate the variation in approaches, CFinder and OCG are applied to the karate network. As can be seen in the Section 5.4, nodes that are allocated more than one module can vary depending on the method. Despite this, all the results shown in Figure 5.3 have the basic form of two modules, with some nodes belonging to both modules. Although in the case of OverMod this is because the hard partition has been restricted to two modules, it also makes sense as the karate network is known to have two modules as its community structure is a result of a dispute between two senior members of the club which led to the original club becoming two separate clubs. CFinder is run on the network with default parameter values and values of k tested are 3, 4, and 5. Immediately, the difference between the results of CFinder and the results reported in the literature is that CFinder fails to cluster all nodes in the network, i.e. some nodes do not belong to any modules. This is a well known property of the CFinder method. In summary, the results are as follows. For $k = 3$, CFinder detects 3 communities, with 2 nodes belonging to two modules (nodes 1 and 32) and 2 nodes belonging to zero modules. For $k = 4$, CFinder detects 3 communities, 2 multi-clustered nodes (nodes 33 and 34) and leaves 22 nodes un-clustered. Finally for $k = 5$, CFinder detects only 1 module, leaves 28 nodes belonging to no modules and since there is only 1 community, of course, there are no overlapping nodes. OCG detects a cover of 21 overlapping modules with 11 multi-clustered nodes: nodes 1, 2, 3, 4, 6, 7, 24, 30, 32, 33, 34. The multi-clustered nodes belong to between two and 12 modules.

The results of OCG and CFinder clearly differ greatly between themselves, and even more so from the results from the literature reported in Section 5.4. In particular,

k	**Modules**	**Not clustered**	**Mono-clustered**	**Multi-clustered**	**2**	**4**
3	25	719	87	5	4	1
4	3	799	12	0	-	-
5	0	811	0	0	-	-
6	0	811	0	0	-	-

TABLE 5.4: CFinder rat PPI network results: number of modules in the soft partition, number of nodes not assigned a module, number of nodes assigned a single module, number of nodes assigned multiple modules, breakdown of the number of nodes the multi-clustered nodes belong to.

CFinder leaves some nodes un-clustered, whereas all other methods considered cluster all nodes in the network. Equally, one could say that OCG finds too many modules since it is known that in most other cases the karate network is found to have between 2 and 4 communities. Furthermore, it could be argued that a member of the karate club belonging to 12 modules is not realistic. However this analysis serves to reinforce that the aim is not to make a strict comparison between methods as the variation between results makes this almost impossible, but merely to highlight differences and similarities between various methodologies and to look for some level of general agreement. The true evaluation of a method is in its ability to assign relevant nodes to multiple modules, which will be looked at in the following section. Now, the overlapping community structure of the two PPI networks, as detected by CFinder and OCG, is now investigated.

CFinder was applied to the main connected components of both PPI networks with default parameter values. For the rat PPI network, $3 \leq k \leq 6$ was tested and for the human PPI network, $3 \leq k \leq 9$. Tables 5.4 and 5.5 give a summary of the results for rat and human respectively. In both cases, a large proportion of the nodes are not assigned a module and the choice of parameter k can clearly considerably change the soft partition detected. For the rat PPI network, CFinder detects multi-clustered nodes only when $k = 3$ and moreover, 719 proteins out of 811 are not clustered. For the human PPI network, CFinder detects multi-clustered nodes for $3 \leq k \leq 6$, although the number of multi-clustered nodes decreases rapidly as k increases. Furthermore, for all k, over half of the nodes are not assigned a module. This is a well-known property of the CFinder method, which in some circumstances may not prove a disadvantage. However it is desirable to make, to some extent, a comparison between methods and it is felt that a comparison with CFinder would not be valuable with such a large degree of the networks remaining unclustered. It is therefore concluded that there is no scope in this study for further investigating the CFinder results.

k	**Modules**	**Not clustered**	**Mono-clustered**	**Multi-clustered**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
3	364	3313	2370	477	369	79	17	10	2	0	0
4	126	5352	635	173	118	33	12	5	3	1	1
5	31	5893	239	28	19	6	2	1	0	0	0
6	8	6063	88	9	9	0	0	0	0	0	0
7	6	6092	68	0	-	-	-	-	-	-	-
8	4	6123	37	0	-	-	-	-	-	-	-
9	1	6151	9	0	-	-	-	-	-	-	-

TABLE 5.5: CFinder human PPI network results: number of modules in the soft partition, number of nodes not assigned a module, number of nodes assigned a single module, number of nodes assigned multiple modules, breakdown of the number of nodes the multi-clustered nodes belong to.

Similarly, OCG is run on the main components of the PPI networks. For the rat PPI network, OCG detects 510 modules with 146 multi-clustered nodes. Overall this overlapping community structure is very different from the soft partition found by running iMod followed by OverMod, which as determined by the iMod hard partition, has 22 modules. The results also differ greatly by the number of modules the multi-clustered nodes belong to. For OCG, multi-clustered nodes belong to between 2 to 180 modules, whereas for OverMod the largest range is from 2 to 8 modules for $r = 0.1$. This result reflects the fact that OCG detects a much larger number of communities than either of the methods used to find hard partitions. For the human PPI network, OCG detects a cover with 393 modules and 2104 multi-clustered proteins. Again the modular structure deviates significantly from soft partition found by Louvain followed by OverMod, which has 19 modules. The multi-clustered nodes belong to between 2 and 53 modules, again very different to the multi-clustered nodes found by OverMod (2 to 14 modules). However, unlike CFinder, OCG allocates at least one module to every node in the network and is therefore more comparable with OverMod.

As said previously the CFinder results deviate too far from those of OverMod and OCG and therefore are not included in the method comparison. For the rat PPI network, OCG multi-clusters almost double the number of proteins multi-clustered by OverMod (146 and 77 for $r = 0.1$ respectively), conversely, OverMod multi-clusters more than double the number multi-clustered by OCG in the human PPI network. Here, it is noted that nodes multi-clustered by OverMod are dependent on the hard partition and therefore the method used to find the hard partition. If the conversion procedure approach is taken then the user must chose a method to detect the hard partition that they feel is reliable or suitable to their needs. In this case, modularity optimisation is selected, a method

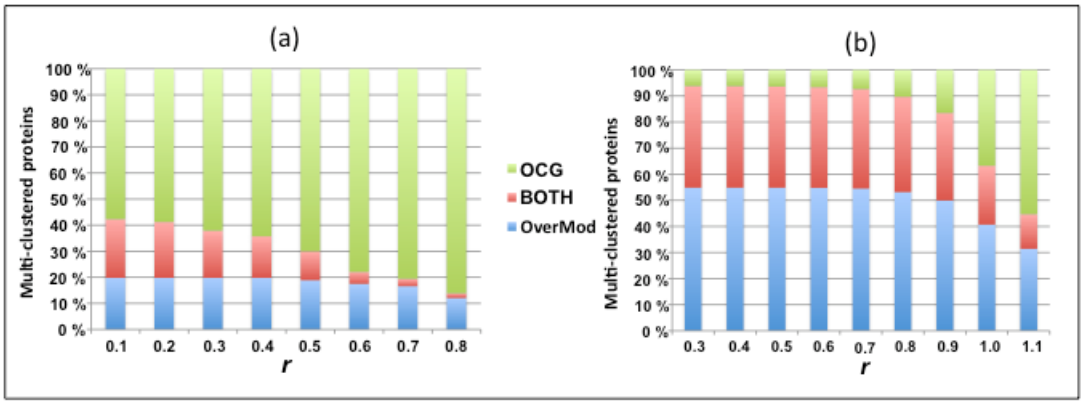

FIGURE 5.8: Comparison of multi-clustered nodes found by OCG and OverMod for (a) the rat PPI network, and (b) the human PPI network. For each network, for each value of parameter r in the figure, the set of nodes multi-clustered by OverMod are always compared with the same set of nodes detected by OCG.

that has been employed by many disjoint community structure detection methods and which has been shown to find relevant solutions in bioinformatics applications.

Returning to the comparison of methods, Figure 5.8 shows the numbers of proteins that are multi-clustered by both methods and uniquely by each method. In each case there is a good level of agreement between the two methods. However, less importance is put on directly comparing methods in terms of the nodes they find to be multi-clustered and more on what is the nature of the multi-clustered nodes and do results make sense in a biological context. Properties of multi-clustered nodes that differentiate them from mono-clustered nodes are discussed in the next section.

5.5.4 Evaluation of the multi-clustered proteins

If nodes that belong to more than one module are interpreted as bridges or connectors between functional units one expects them to exhibit properties that reflect such responsibilities. In this section, node degree and Gene Ontology (GO) annotations [23] are employed as measures of a node's structural and functional importance. It is investigated whether, given an appropriate hard partition, OverMod can multi-cluster proteins that are distinguishable from mono-clustered proteins in terms of these properties. Furthermore, these descriptive features are used to indicate an appropriate range of values of parameter r for a particular network. Finally, focus is turned to several strongly

r	**Multi-clustered**	**Mono-clustered**	**p-value**
0.1	7.81	1.76	3.05E-45
0.2	5.12	2.05	3.28E-43
0.3	4.07	2.17	1.77E-37
0.4	3.85	2.20	7.03E-34
0.5	3.30	2.26	4.36E-25
0.6	2.77	2.31	3.81E-15
0.7	2.47	2.33	4.06E-12
0.8	2.22	2.34	2.09E-07
OCG	7.17	1.27	7.26E-100

TABLE 5.6: Average node degrees of multi- and mono-clustered nodes detected by OverMod and OCG for the rat PPI network.

multi-clustered proteins to determine whether the proposed methodologies can indeed highlight some proteins known to be functionally important.

5.5.4.1 Connectivity of multi-clustered proteins

The average node degree of multi-clustered proteins is compared with that of mono-clustered proteins found by OverMod and OCG for both networks. The population means are determined to be statistically significantly different or not using the Mann-Whitney-Wilcoxon U test as implemented in the statistical computing environment R [164]. Results for the rat PPI network are shown in Table 5.6 for OverMod and OCG. The MINLP in OverMod was solved for $0.1 \leq r \leq 0.8$. The average node degree is higher for multi-clustered proteins than for the mono-clustered proteins for $0.1 \leq r \leq 0.7$ and this difference is statistically significant for the same range, where a p-value < 0.01 is significant. At $r = 0.8$ the converse is true; mono-clustered proteins have a significantly higher average degree than multi-clustered proteins. For OCG, the difference between average node degree of multi-clustered and mono-clustered is also significant.

For the human PPI network, the results are shown in Table 5.7. For OverMod, the average node degree for the multi-clustered proteins is greater than for the mono-clustered nodes for $0.3 \leq r \leq 1$ and this difference is statistically significant for the same range. At $r = 1.1$, the average degree of the mono-clustered proteins is significantly higher than that of the multi-clustered proteins. OCG also finds multi-clustered proteins that have a significantly higher average connectivity than mono-clustered proteins.

r	**Multi-clustered**	**Mono-clustered**	**p-value**
0.3	9.71	3.12	$<$ 2.2E-16
0.4	9.71	3.11	$<$ 2.2E-16
0.5	9.71	3.14	$<$ 2.2E-16
0.6	9.71	3.21	9.20E-299
0.7	9.75	3.32	3.41E-279
0.8	9.68	4.24	5.35E-203
0.9	9.64	5.35	1.61E-107
1	8.38	7.46	3.52E-4
1.1	6.22	8.25	6.85E-11
OCG	14.52	4.31	$<$ 2.2E-16

TABLE 5.7: Average node degrees of multi- and mono-clustered nodes detected by OverMod and OCG for the human PPI network.

For both networks, OverMod assigns proteins to more than one module that have on average a higher connectivity than those belonging to only one module for a finite range of values of parameter r. This idea that multi-clustered proteins are more highly connected than mono-clustered proteins is intuitive, since if proteins lying in the overlapping sections play a connector role by interacting with two or more modules, it is reasonable that they are more likely to interact with more partners compared to isolated nodes. It is noted here that, although the inclusion of the parameter r is advantageous as it offers the user greater flexibility, it is necessary to determine a reasonable range of values for each network. Node degree can be used as an indicator for r if it is assumed that nodes belonging to more than one community should have more interactions than those that do not as this can act as an indicator of structural importance. Therefore, here it has been shown that in terms of node degree, the range of values of r that detect significant differences between multi- and mono-clustered nodes is finite and that for the rat network a reasonable range of values is between 0.1 and 0.7 inclusive and similarly for the human network, 0.3 to 1 inclusive.

5.5.4.2 Multi-functionality of multi-clustered proteins

Where node degree offers a topological measure for illustrating the structural importance of multi-clustered proteins, GO annotations can act as a descriptive feature based on functionality. The Gene Ontology (GO) project offers consistent descriptions of gene products in three structured controlled vocabularies (ontologies) in terms of associated biological processes, cellular components and molecular functions. Each ontology is

structured as a directed acyclic graph, with general terms at the root of the graph, with annotations becoming more specific as one moves down the graph. For example, a broad molecular function term is "catalytic activity", whereas a more specific term on the same branch as catalytic activity is "adenylate cyclase activity".

As seen in [30], and in line with the interpretation in this study of multi-clustered nodes as bridges between modules, one would expect their multi-functionality to be reflected in the number of GO annotations. To test this hypothesis, the average number of GO terms annotated to multi- and mono-clustered proteins is compared to determine whether proteins belonging to more than one module are associated with a significantly higher number of functions than those belonging to only one module. The combined total of all three categories of GO terms (ALL GO) as well as each aspect individually (molecular function, MF, biological process, BP and cellular compartment, CC) are considered. GO annotations were downloaded for rat and human from [7] and [8] respectively. The annotations were then filtered to remove any redundant parent terms and each protein was mapped to its GO terms, where possible. The average number of GO terms (ALL GO, MF, BP and CC) for the set of multi-clustered proteins and the set of mono-clustered proteins was then calculated (excluding genes with no GO annotations). The results are shown for rat and human in Table 5.8 and 5.10 respectively.

For the rat PPI network, multi-clustered proteins have a higher average number of annotations for ALL GO, MF, BP and CC for $0.1 \leq r \leq 0.9$. The difference is statistically significant for $0.1 \leq r \leq 0.7$ for ALL GO, MF and BP $0.1 \leq r \leq 0.6$ for CC, according to the Mann-Whitney-Wilcoxon U test, as shown in Table 5.9. For OCG, the multi-clustered proteins have a significantly higher average number of annotations for ALL GO, MF and CC, but not for BP.

Similarly for the human PPI network, OverMod finds multi-clustered proteins with a higher average number of GO annotations for ALL GO, MF, BP and CC when $0.3 \leq r \leq 1.1$, shown in Table 5.10. Table 5.11 shows that the difference in average number of annotation is significant for $0.3 \leq r \leq 1.0$. OCG also detects multi-clustered proteins with a significantly higher average number of annotations for all four categories tested.

The above results indicate that in general, proteins lying in the overlap between communities possess a wider functional repertoire than those belonging to only one community. Here, the number of GO annotations of a given protein is taken to indicate a level of functional importance, therefore these results support to some extent the hypothesis made in this study that multi-clustered proteins play a crucial functional role in the

	Multi-clustered				**Mono-clustered**			
r	**ALL GO**	**MF**	**BP**	**CC**	**ALL GO**	**MF**	**BP**	**CC**
0.1	29.03	6.31	17.43	6.10	18.41	4.34	10.63	4.30
0.2	29.23	6.36	17.59	6.11	18.39	4.34	10.61	4.30
0.3	29.41	6.28	17.81	6.14	18.49	4.36	10.67	4.31
0.4	29.32	6.22	17.93	6.05	18.56	4.38	10.71	4.33
0.5	28.80	6.34	17.61	5.78	18.76	4.40	10.85	4.38
0.6	26.78	6.03	16.21	5.61	19.07	4.45	11.06	4.41
0.7	28.16	6.37	17.41	5.71	19.06	4.45	11.05	4.42
0.8	28.25	6.42	17.83	6.10	19.19	4.48	11.14	4.43
0.9	27.61	6.33	17.19	6.00	19.23	4.48	11.17	4.43
OCG	29.45	6.50	17.43	6.13	17.08	4.06	17.08	4.08

TABLE 5.8: Average number of GO annotations for rat PPI Network.

r	**All GO**	**MF**	**BP**	**CC**
0.1	**1.76E-05**	**5.49E-06**	**1.26E-04**	**2.30E-05**
0.2	**1.17E-05**	**2.99E-06**	**9.19E-05**	**1.81E-05**
0.3	**5.18E-06**	**4.79E-06**	**3.57E-05**	**9.26E-06**
0.4	**2.00E-05**	**1.51E-05**	**5.62E-05**	**4.53E-05**
0.5	**1.41E-04**	**1.07E-04**	**2.26E-04**	**3.43E-04**
0.6	**6.28E-03**	**1.46E-03**	**6.02E-03**	**7.55E-03**
0.7	**8.35E-03**	**9.07E-04**	**6.16E-03**	1.38E-02
0.8	1.04E-01	2.74E-02	6.71E-02	3.87E-02
0.9	9.54E-02	4.80E-02	6.01E-02	4.90E-02
OCG	**6.85E-12**	**4.52E-12**	2.43E-01	**2.93E-07**

TABLE 5.9: Significance test results for the difference in population mean for number of GO terms between overlapping and non-overlapping genes for the rat PPI network. Significant values are shown in bold.

	Multi-clustered				**Mono-clustered**			
r	**ALL GO**	**MF**	**BP**	**CC**	**ALL GO**	**MF**	**BP**	**CC**
0.3	14.64	3.36	8.65	3.31	11.42	2.80	6.92	2.59
0.4	14.64	3.36	8.65	3.31	11.41	2.80	6.91	2.59
0.5	14.64	3.37	8.65	3.31	11.42	2.80	6.91	2.59
0.6	14.66	3.37	8.67	3.31	11.40	2.80	6.89	2.60
0.7	14.68	3.37	8.69	3.31	11.46	2.81	6.89	2.62
0.8	14.78	3.37	8.74	3.33	11.67	2.88	7.02	2.66
0.9	15.07	3.40	8.93	3.39	11.89	2.95	7.08	2.72
1.0	15.38	3.41	9.28	3.39	12.83	3.11	7.54	2.96
1.1	14.14	3.24	8.47	3.19	13.73	3.22	8.15	3.11
OCG	16.60	3.61	9.86	3.64	12.21	2.99	7.22	2.83

TABLE 5.10: Average number of GO annotations for human PPI Network.

r	All GO	MF	BP	CC
0.3	**7.03E-26**	**2.27E-14**	**1.38E-14**	**1.25E-22**
0.4	**6.17E-26**	**2.44E-14**	**1.04E-14**	**1.44E-22**
0.5	**9.79E-26**	**2.12E-14**	**1.29E-14**	**2.72E-22**
0.6	**1.98E-26**	**1.14E-14**	**3.25E-15**	**1.08E-21**
0.7	**8.87E-26**	**3.98E-14**	**2.52E-15**	**9.41E-22**
0.8	**4.29E-25**	**2.24E-11**	**2.75E-14**	**1.41E-22**
0.9	**1.78E-27**	**1.17E-11**	**2.83E-17**	**2.21E-22**
1.0	**2.74E-12**	**6.26E-06**	**8.51E-10**	**3.89E-07**
1.1	8.63E-01	9.46E-01	7.60E-01	9.15E-01
OCG	**2.48E-35**	**3.32E-18**	**4.36E-19**	**4.57E-30**

TABLE 5.11: Significance test results for the difference in population mean for number of GO terms between multi- and mono-clustered proteins for the human PPI network. Significant values are shown in bold.

entire system. Similar to node degree, the assumption of multi-clustered proteins being more multi-functional than mono-clustered proteins is used to let the number of GO terms act as an indicator of an appropriate range of values for parameter r. In conclusion, according to this analysis, to be true for all aspects of GO (ALL GO, MF, BP and CC), an appropriate range for the Rat PPI network is between 0.1 and 0.6 inclusive and 0.3 to 1.0 for the Human PPI network.

5.5.4.3 Strongly multi-clustered proteins

It has been shown that the proteins assigned to multiple modules by OverMod, have a high average degree and a high average number of biological annotations. Here, a few specific examples are identified that help to illustrate the methods ability to identify proteins that are known to be important in some way, e.g. associated with a certain disease. The most strongly multi-clustered nodes for each value of r are defined as the nodes that belong to the largest number of modules. Here, the strongly multi-clustered proteins in the human and rat PPI networks are explored.

Becker et al. [30] found that the top ten most multi-clustered proteins by OCG in the human PPI network were involved in general regulatory functions, e.g. ubiquitination, regulation of transcription and signalling, functions that are involved in multiple biological processes. Furthermore they found that the multi-clustered proteins are enriched for proteins involved in cancer. Table 5.12 shows the top ten most multi-clustered by OCG reported in [30] and the number of modules OverMod assigns them to for each value of r. For $0.3 \leq r \leq 0.8$ OverMod multi-clusters all proteins in the top 10, for

	r							
OCG Top Ten	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
UBQL4	16	15	11	10	8	3	1	1
P53	13	13	13	12	10	6	4	2
SMAD2	15	15	12	12	10	7	3	2
EP300	10	10	10	10	9	7	3	1
SMAD3	14	14	12	12	10	7	2	2
SMAD9	15	15	13	11	8	3	2	1
TRAF2	11	11	9	7	6	2	2	2
EGFR	12	12	10	9	9	5	5	2
CBP	8	8	8	8	7	5	3	1
TGFR1	17	15	12	11	9	7	3	1

TABLE 5.12: The top ten most strongly multi-clustered proteins by OCG as reported in [30].

$r = 0.9$, OverMod multi-clusters all but one of the top 10 and for $r = 1.0$, OverMod multi-clusters 5 of the top 10.

The proteins most strongly multi-clustered by OverMod for the human PPI network are now discussed. For $0.3 \leq r \leq 1.0$, the top few proteins belonging to the largest number of modules are selected. For example, at $r = 0.3$, the proteins belonging to 17, 16 and 15 modules are chosen as the most strongly multi-clustered and for $r = 0.4$, the proteins belonging to 15 and 16 modules are selected. For $0.3 \leq r \leq 1.0$, all strongly multi-clustered proteins are combined and Table 5.13 shows the distribution across the range of r. Those that were found in the set of most strongly multi-clustered for only one value of r were removed (this includes all those proteins belonging to 4 modules at $r = 1.0$). Some of the proteins in Table 5.13 are discussed below.

Several of the most strongly multi-clustered proteins in the human PPI network are associated with cancer, in accordance with the results found in [30]. A closer look is taken at these proteins below. Much of the information regarding the strongly multi-clustered proteins comes from the UniProt database [199] unless otherwise stated.

- 14-3-3 protein gamma (1433G_HUMAN), encoded by the YWHAG gene, is implicated in the regulation of many general and specialized signalling pathways. Although not directly associated with cancer, it has been shown to interact with RAF1, which as noted above, is a proto-oncogene.

r	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
UBIQ_HUMAN	16	16	14	13	11	0	0
1433G_HUMAN	15	15	13	12	10	0	0
HD_HUMAN	15	15	13	13	11	0	0
SMAD2_HUMAN	15	15	0	12	10	7	0
CDC2_HUMAN	16	16	14	14	11	7	6
SMAD9_HUMAN	15	15	13	0	0	0	0
ATX1_HUMAN	17	15	0	0	0	0	0
A4_HUMAN	0	0	0	12	10	7	6
RAF1_HUMAN	0	0	13	13	11	7	0
SMAD4_HUMAN	0	0	13	12	10	7	0
EWS_HUMAN	15	0	0	0	0	0	0
TGFR1_HUMAN	17	15	0	0	0	7	0
ANDR_HUMAN	0	0	0	12	10	7	0
BRCA1_HUMAN	0	0	0	12	11	7	0
CASP3_HUMAN	0	0	0	0	10	7	6
CBL_HUMAN	0	0	13	12	10	0	0
GBLP_HUMAN	0	0	0	0	10	7	6
P53_HUMAN	0	0	13	12	10	0	0
RB_HUMAN	0	0	13	13	10	0	0
SMAD3_HUMAN	0	0	0	12	10	7	0
UBQL4_HUMAN	16	15	0	0	0	0	0
ARRB2_HUMAN	0	0	0	0	0	7	7
CTNB1_HUMAN	0	0	0	12	0	7	0
DYL1_HUMAN	0	0	0	0	10	7	0
KGP1B_HUMAN	0	0	0	0	0	7	6
MDM2_HUMAN	0	0	0	0	10	7	0
SKP2_HUMAN	0	0	0	0	0	7	6
TBA4A_HUMAN	0	0	0	0	0	7	6
TBB5_HUMAN	0	0	0	0	0	7	6
TF65_HUMAN	0	0	0	0	10	7	0
YAP1_HUMAN	0	0	0	0	0	7	7
ZFYV9_HUMAN	0	0	0	12	10	0	0

TABLE 5.13: The most strongly multi-clustered proteins in the human PPI network multi-clustered by OverMod.

- Mothers against decapentaplegic homolog 2 (SMAD2_HUMAN), encoded by the SMAD2 gene, is a member of the SMAD family which has been shown to play a tumour suppressor role in prostate cancer [221].

- Cyclin-dependent kinase 1 (CDC2_HUMAN) is a member of the cyclin-dependent kinase (CDK) family, a group of multifunctional enzymes involved in regulation of cell cycle. Loss of cell cycle control is a hallmark of cancer and consequently CDK1 has been shown to be an effective therapeutic target for inhibitors in cancer treatment [209].

- The mothers against decapentaplegic homolog 9 protein (SMAD9_HUMAN), encoded by the SMAD9 gene, is another member of the SMAD family of proteins. SMAD9 interacts with the tumour suppressor gene SMAD4, another of the most strongly multi-clustered proteins mentioned below.

- The RAF proto-oncogene serine/threonine-protein kinase (RAF1_HUMAN), encoded by the RAF1 gene, has been suggested as a potential target for the therapeutic intervention of the onset of K-Ras oncogene-driven non-small cell lung carcinoma (NSCLC) [33].

- The mothers against decapentaplegic homolog 4 protein (SMAD4_HUMAN), encoded by the SMAD4 gene, which is a tumour suppressor gene found to be mutated in many cancers, including pancreatic and colorectal cancer.

- The RNA-binding protein EWS (EWS_HUMAN), encoded by the EWSR1 gene, is involved in various cellular processes, including gene expression, cell signalling, and RNA processing and transport. Mutations in the EWSR1 gene are known to cause Ewing sarcoma and various other tumours.

- The androgen receptor (ANDR_HUMAN), encoded by the AR gene, is associated with prostate cancer and has been shown that the inhibition of AR activity may delay prostate cancer progression [89].

- The breast cancer type 1 susceptibility protein (BRCA1_HUMAN), encoded by the BRCA1 gene, is responsible for DNA repair. Defects in the gene are associated with an increase risk of breast cancer and also ovarian and pancreatic cancers. The BRCA-FA pathway, of which the BRCA1 gene is a component, has been shown to be a potential target for anti-tumour drugs [124].

- The E3 ubiquitin-protein ligase CBL (CBL_HUMAN), encoded by the CBL gene, functions as a negative regulator of many signalling pathways involved in cell signalling and protein ubiquitination. CBL is a proto-oncogene associated with a number of human cancers, in particular acute myeloid leukemia [38].

- p53 (P53_HUMAN) is a tumour suppressor protein, encoded by the TP53 gene, which regulates the cell cycle and is well-known to be heavily involved in cancer. TP53 is mutated or inactivated in over 50% of cancers [155]. It has been shown that reintroduction of functional p53 into tumours has a therapeutic benefit [190].

- The retinoblastoma-associated protein (RB_HUMAN), encoded by the RB1 gene, is a regulator of cell division that acts and a tumour suppressor. RB1 is mutated in several cancers including childhood cancer retinoblastoma, bladder cancer and osteogenic sarcoma. The components of the RB pathway have been proposed as potential targets for treating cancer [100].

- The mothers against decapentaplegic homolog 3 protein (SMAD3_HUMAN) is encoded by the SMAD3 gene. Defects to this gene are associated with colorectal cancer. Furthermore, SMAD3 has been shown to be over expressed in prostate cancer [129].

- The protein, Catenin beta-1 (CTNB1_HUMAN), is encoded by the CTNNB1 gene. Defects in this gene are associated with several cancers including colorectal and ovarian.

It is noted here that it is unsurprising that both p53 and BRAC1 are strongly multi-clustered as there will be a bias, since they have both been well-explored and therefore it is likely that more of their interaction partners are known.

Furthermore, the Huntingtin protein (HD_HUMAN), which is encoded by the HTT gene, is responsible for causing Huntington's disease, a neurodegenerative genetic disorder where sufferers experience involuntary movements (chorea), general motor impairment, psychiatric disorders and dementia. Studies have indicated that HTT is a multifunctional protein that plays distinct roles in several biological processes, including synaptic transmission, intracellular transport and neuronal transcription [135].

Two proteins associated with Alzheimer's disease appear in the most strongly multi-clustered set. The Amyloid beta A4 protein (A4_HUMAN) is encoded by the APP gene. Defects of the APP gene are the cause of Alzheimer's disease type 1 (Uniprot).

Regulators of APP expression have been shown to be potential targets for Alzheimer's disease therapeutics [127]. Similarly, Caspase-3 (CASP3_HUMAN), a protein encoded by the CASP3 gene, has been found to be a potential therapeutic target during the early stages of Alzheimer's disease [52].

Finally, as also found in [30], proteins associated with ubiquitination are found to be strongly multi-clustered. Ubiquitination is the process of labelling proteins for degradation through the addition of an ubiquitin molecule to the target protein. This process enables the cell to dispose of misfolded or damaged proteins and control the concentration of essential proteins, and is associated with multiple functions, including cell cycle regulation, apoptosis and immunity and with cancer and neurodegenerative diseases. UBIQ_HUMAN and UBQL4_HUMAN are both found to be among the strongly multi-clustered proteins detected by OverMod. Similarly, although only a few proteins are assigned to more than three modules in the rat PPI network, for r equal to 0.1 and 0.2, the most strongly multi-clustered protein is associated with ubiquitination, belonging to 8 and 5 modules respectively.

Overall, these examples illustrate the ability of OverMod to recover proteins known to play important roles in a variety of disease and regulatory processes and reinforces the potential of the proposed method in pin pointing important nodes in a network. These organisms are well-annotated therefore much is already known about many of the multi-clustered proteins that are found, however, the real ability of the method could be demonstrated when analysing less well annotated organisms.

5.5.5 PPI network analysis discussion and conclusions

In this section, OverMod was applied to the PPI networks of two well-annotated organisms, rat and human, to assess the method's ability to extract biologically meaningful results. More specifically, the aim was to determine whether OverMod could multi-cluster proteins with characteristics that indicate a relevant connector role in the network. OverMod is the second stage in a two-part procedure for detecting the overlapping community structure of a network, where the first stage involves detecting a hard partition. Here modularity optimisation methods were used to partition the PPI networks into disjoint communities. It was then the turn of OverMod to convert the disjoint communities into overlapping communities. The MINLP was solved for a range of values of r and the results evaluated according to the hypothesis that multi-clustered nodes are the

connectors between functional units (modules) and should therefore exhibit properties conducive to this role.

It was found that the multi-clustered proteins had a higher average degree and a higher average number of GO annotations than mono-clustered proteins. Furthermore, these features were used to suggest a range of values for the overlapping parameter, r. If high connectivity and high multi-functionality are assumed to be desirable characteristics of multi-clustered proteins then a range of values of r can be chosen where they possess these properties. Additionally, several strongly multi-clustered proteins were selected for further discussion. Many of the most strongly multi-clustered were found to be associated with cancer, similar to findings in [30]. These results illustrate the ability of OverMod to recover proteins known to have important properties and therefore indicate the methods potential to predict functionally important proteins or genes in biological systems.

Comparisons were made with CFinder [159] and the OCG method [30]. As is discussed above, CFinder leaves a large proportion of the network un-clustered and therefore the results were not fully analysed. For both networks, a reasonably large number of proteins were multi-clustered by both OverMod and OCG. Furthermore, the multi-clustered proteins found by OCG were also shown to exhibit high connectivity and multi-functionality. With the exception of one case, where OCG failed to detect multi-clustered proteins with a higher average number of BP annotations than mono-clustered proteins in the rat PPI network. However, the real difference in the results is that OCG detects a much larger number of overlapping modules than any of the modularity optimisation methods used here. For example, for the karate network, OCG finds a soft partition of 21 overlapping modules, which is not reasonable for this network. Therefore, at this point one must make the choice as to which methodology better addresses the needs of the problem being investigated. In this study, the overlapping community detection problem is viewed as an extension of previous clustering methods, therefore the proposed two-stage approach is more appropriate than CFinder or OCG in this context.

5.6 Discussion and conclusions

In this chapter, a mixed integer non-linear programming (MINLP) model, known as OverMod, was proposed to convert disjoint communities to overlapping communities.

OverMod is the second stage in a two-stage approach, where the first stage involves detecting a hard partition of a network, which is then converted to a soft partition. The method's performance was first evaluated on the Zachary karate network. A comparison was made with results reported in the literature, showing that even on such a small network, the large variation in existing methodologies is reflected in the results. A deeper analysis of the nature of the multi-clustered nodes followed with applications on the rat and human PPI networks. Results showed that multi-clustered proteins were on average more highly connected and more highly multi-functional than mono-clustered proteins. Moreover many of the most 'strongly' multi-clustered proteins detected were known to be associated with disease, and in particular cancer. These results corroborate to some extent the idea that multi-clustered proteins play strategically important roles in the network, allowing functional units (modules) to interact and regulate functions required by the system. Future work will investigate other features of the multi-clustered nodes that reinforce their connecting role. For example, determining whether multi-clustered proteins contain more domains than mono-clustered proteins. In addition, for the human PPI network, it will be investigated whether multi-clustered proteins are enriched for druggable targets according to the druggable genome [177] as has been done in [223]. This will again help to confirm whether the methodology presented here can identify important nodes and has the potential to contribute to detecting drug targets.

The results of applying hard partitioning methods followed by OverMod to the PPI networks were compared with two other overlapping community structure detection methods, CFinder [159] and OCG [30]. The CFinder results were not fully analysed as the method fails to cluster a large proportion of the nodes, a well-known property of the method. Therefore it was felt that a comparison with the other methods would be unfair. It was also felt that CFinder does not tackle the overlapping community structure detection problem in a way that is satisfactory to the interpretation of the problem in the context of this study. One of the aims of community structure detection is to associate nodes with unknown functions or properties with nodes of known functions or properties. Of course, it doesn't make sense to associate nodes if there really is no connection, however it is possibly beneficial to associate the node with the most likely functional module or modules in order to generate some hypothesis regarding the nature of a node. In this sense it may be better to be inclusive rather than exclusive and therefore it is felt that CFinder disregards too much of the information about the system provided by the network.

Similarly to OverMod, OCG finds multi-clustered proteins that are more highly connected and are associated with more GO annotations than mono-clustered proteins. In terms of significance (p-values), the results do not differ greatly between both methods; the main difference between the results lies in underlying modular structure. For both PPI networks, the OCG soft partition comprises over 20 times more modules than the hard partitions used by OverMod. This is down to differences in fundamental methodology, in particular, OCG starts its agglomerative procedure with an initial cover of the network comprising a large number of classes/modules, which are subsequently fused until one of three stopping criteria is achieved. Consequently, if the criteria is stopped after a relatively few number of iterations, the resulting number of modules is high. The number of overlapping modules in the final cover of the network detected by OverMod on the other hand is dependent on the method used to find the hard partition. In both examples in this study, methods based on modularity optimisation are used, a well-recognised approach to community structure detection, employed by many methods. However, the debate about which is the most realistic partition of the network, is beyond the scope of this chapter and the aim here is to simply show that the proposed method can identify structurally and functionally relevant nodes, according to the application design adopted here. Nonetheless, it can be said that both are valid methods, and choice of method, really depends on the specific experimental needs of the user.

An advantage of OverMod is that it offers a more informative modelling framework by calculating belonging coefficients for each node-module association. Both OCG and CFinder do not provide this deeper level of description of the system. Future work will include (i) analysing how the belonging coefficients of a multi-clustered gene are distributed between communities and (ii) identifying genes that are more equally spread among functional communities than others.

Further benefits of OverMod arise due to the flexible modelling framework. This flexibility comes in part from the inclusion of two parameters: r to control the size of the overlap and K to filter the results according to the strength of belonging of a node to its dominant module. Choice of parameters depends on the specific application or the user's requirements, however for parameter r, it is shown how the connectivity and multi-functionality of proteins indicate an appropriate range of values for protein interaction networks. The effect of parameter K is not thoroughly investigated here, however this will be covered in future work. In general, parameter values should be explored in specific applications. These parameters offer the user more power over their analysis, which is important when the problem statement is not well defined.

Further control is offered by the fact that the user can choose any hard partitioning method, allowing them to use a method that they are familiar with or that they feel is reliable. This flexibility is illustrated in this chapter as different methods were chosen for each network: WeiMod for the karate network, iMod for the rat PPI network and Louvain for the human PPI network. Choice of method was based on network size and the value of modularity achieved by each method. Furthermore, due to the nature of mathematical programming models, additional constraints and parameters can be easily implemented, again leading to the possibility of more accurate and detailed network representations. For example, prior knowledge of a system could be used, such that nodes with similar functional annotations could be constrained to be in the same community. In terms of methodology, introducing symmetry constraints to as has been done in previous models [219] may improve the efficiency of OverMod. Further improvements to efficiency may come from using alternative solvers, as described in Chapter 4.

Overall, the detection of overlapping modules allows the intersection of functional modules to be investigated, the nodes which connect the functional units to be identified and helps to provide a greater understanding of the underlying mechanisms of the system under study. It has been shown that the proposed method has the ability to pin point important proteins/genes, demonstrating its potential in future bioinformatics applications. The applicability of the procedure outlined in this chapter is now tested further in an exploratory analysis of a fungal pathogen in the following chapter.

Chapter 6

Exploration of the community structure of an integrated network of the fungal pathogen *Fusarium graminearum*

Uncovering the densely connected communities of a biological network reveals a high level view of its constituent functional units. At the same time, investigating the overlapping sections between communities offers insights into the roles of individual genes in the context of the entire network. Both facets of community structure detection contribute towards a better understanding of the underlying organisation of the biological system. In this chapter, these analytical procedures are applied to an integrated network of the major fungal pathogen of many cereal crops *Fusarium graminearum*. Infections by *Fusarium* have a significant impact on grain yield and quality and therefore a deeper understanding of the nature of the pathogen is desired by the agricultural industry. Here, the disjoint and overlapping community structure of the network generated from sequence, protein interaction and co-expression data is explored in an attempt to link topological and functional features. More specifically, the functional coherence of communities, properties of multi-clustered genes and the relationship between virulence-associated genes and community structure are all examined. This exploratory study is a first step in deciphering the underlying mechanisms of the pathogen, and moreover,

it highlights the potential contributions of the methodology described in this thesis in future bioinformatics applications.

6.1 Introduction

In addition to genome sequence data, a large amount of multiple complimentary types of biological data is now available for many organisms, such as gene expression, protein interactions and phenotypic information. Data from various sources can be integrated to build complex networks where nodes are proteins or other biological entities and edges capture the intricate associations between them [113, 114]. Integrated networks can provide a framework to explore topological-functional relationships in biological systems. Exploring the community structure of such integrated networks can identify functionally coherent units and give insight into higher levels of biological organisation [83, 125]. As has been discussed in the previous chapter, proteins can take part in multiple processes [103] and therefore a better description of the underlying functional mechanisms may be provided by considering the intersections of communities.

The Ascomycete fungus *Fusarium graminearum* is a major pathogen of wheat, causing *Fusarium* ear blight, *Fusarium* head blight or *Fusarium* head scab disease [55, 78]. See Figure 6.1. The pathogen also infects numerous other cereal crops, including maize, barley, triticale, rice and oats [78]. Floral infections by *Fusarium* can have a significant impact on grain yield and quality. In addition, infection by the fungus leads to contamination by various mycotoxins including deoxynivalenol (DON), making the grain harmful for human and animal consumption. As wheat constitutes 32% of global cereal production and provides 20% of the worlds calorific intake [9], research into the disease process of *Fusarium* is important due to the potential implications in the agricultural industry. To facilitate this, the complete genome sequence (with 13,718 protein coding genes) of *Fusarium* has been determined [51] and additionally data on gene expression [214] and predicted protein interactions [229] also exist. The availability of such data has lead to the construction of an integrated network of the pathogen that combines sequence, protein interaction and co-expression data [31].

In contrast to rat and human, the highly characterised organisms studied in Chapter 5, *Fusarium* is much less well annotated and therefore a much more exploratory study is now presented. In this chapter the modular structure of the integrated network for *Fusarium* is explored with the aim of gaining insights into the organisation of the pathogen

FIGURE 6.1: *Fusarium* on wheat. Healthy wheat is to the left, infected wheat is to the right. Image taken from [10].

at the cellular level. To this end, the detection of disjoint and overlapping community structure is employed as a means of investigating topological-functional relationships in the pathogen. Functional coherence of communities, properties of multi-clustered genes, including connectivity, multi-functionality and number of protein domains are examined. Additionally the connection of known and predicted virulence genes to community structure is investigated in an attempt to topologically characterise such genes. It is hoped that this study can represent a first tentative step into the difficult task of understanding the disease process of *Fusarium*. In parallel, the goal is to build on the evidence from Chapter 5 and further demonstrate the ability of the community structure detection based methods presented in this thesis to extract meaningful results in biological networks, leading the way for future bioinformatics applications.

6.2 Methods

The construction of an integrated network for *Fusarium graminearum* is described in [31]. The network was generated using information from sequence similarity, co-expression

No. of nodes	2	3	4	5	6	7	9	10	11	16	8364
No. of components	288	101	23	10	5	3	3	2	2	1	1

TABLE 6.1: Connected components in the integrated network.

and predicted protein interactions (PPI). The sequence similarity network was constructed by carrying out pairwise sequence matches of the proteins in version 3.2 of the *Fusarium graminearum* annotation [11] implemented on a TimeLogic Tera-BLAST (Active Motif Inc., Carlsbad, CA) system with a threshold E-value for bidirectional best hits of 10^{-6}. Co-expression information was obtained from the publicly available set of *Fusarium* expression studies from PLEXdb [214] that used Fusarium Affymetrix GeneChip arrays. Similarity between expression profiles was measured using the weighted Pearson correlation coefficient, according to the method in [154]. PPI information was taken from the predicted core PPI in [229]. Two proteins in the network are linked if any of the following properties are satisfied: (i) a bi-directional sequence similarity BLAST hit has an expected value of less than 10^{-6}, (ii) the Pearson correlation coefficient of the two gene expression profiles has an absolute value greater than 0.88, or (iii) there exists a PPI between the two proteins in the dataset from [229]. Integration of the various data sources was carried out using the Ondex data integration platform [104, 132].

The network comprises 9521 nodes (proteins), 80997 links and is made up of 439 disconnected components. Table 6.1 shows the distribution of sizes of the connected components. The disjoint and overlapping community structure of the network is investigated using methodology that has featured in earlier chapters. The disjoint community structure of the largest connected component of the network is detected using the greedy agglomerative method, Louvain [34]. Louvain is chosen due to the large scale of the network and the low computational cost of the method (for more details see Chapter 2, Section 2.3.2). The disjoint communities are then converted to overlapping communities through the application of OverMod, described in Chapter 5. Values of parameter r, which controls the extent of the overlapping, range from 0.1 to 1.1 inclusive. The results are presented in the following section.

6.3 Results

In this section the results of the analysis of the integrated network of *Fusarium graminearum* are presented. First the disjoint community structure is detected and the functional coherence of the modules assessed. The disjoint communities are transformed to

overlapping communities and like in Chapter 5, the connectivity and multi-functionality of the multi-clustered proteins are compared with those of the mono-clustered proteins. In this chapter however, in addition to node degree and number of GO terms, average number of protein domains of the multi-clustered proteins is also examined. Furthermore, the functional cartography scheme presented in [83] (Chapter 2, Section 2.2.4.2) is employed in the context of overlapping community structure to assign topological roles to multi-clustered proteins, thus providing a deeper level of structural description. Finally the positions of known and predicted pathogenicity-associated genes within modular structure are explored in an attempt to make a connection between pathogenicity and structural network properties.

6.3.1 Disjoint community structure detection

This analysis focuses on the largest connected component of the integrated network, which comprises 8364 nodes and 79931 links, as community structure of smaller components is of limited scope. The main component of the network is partitioned by the Louvain method [34], which detects a partition of 91 disjoint communities with modularity equal to 0.7973. The resultant community structure has an uneven community size distribution, with 89 communities of size < 500 and 2 large communities with 1007 and 1951 nodes respectively. The module size distribution is shown in Figure 6.2. The partition found by Louvain is compared with that of QCUT [176] (Chapter 2, Section 2.3.5). QCUT finds a partition with 53 communities (modularity equal to 0.7665), 51 of which have <500 nodes and two larger communities with 1198 and 2968 nodes (Figure 6.2). This is in agreement with the uneven community structure found by the Louvain method. Based on Louvain finding the larger value of modularity, this hard partition is used in the remainder of the analysis.

The disjoint community structure is illustrated by a meta-view of the partition in Figure 6.3, with (i) the size of communities, (ii) the number of shared nodes across communities in the overlapping community structure (discussed in the following section) and (iii) their functional content. The functional coherence of a community is assessed by the Average Information Content of the Most Informative Common Ancestor set (AIC-MICA) a metric defined in [132]. The information content (IC) of an annotation term is calculated based on how frequently a particular annotation is found in an annotation set of a given species. The annotation set used here is the Gene Ontology (GO) [23]. The AIC-MICA approach takes as input a set of entities (i.e. the genes in a module)

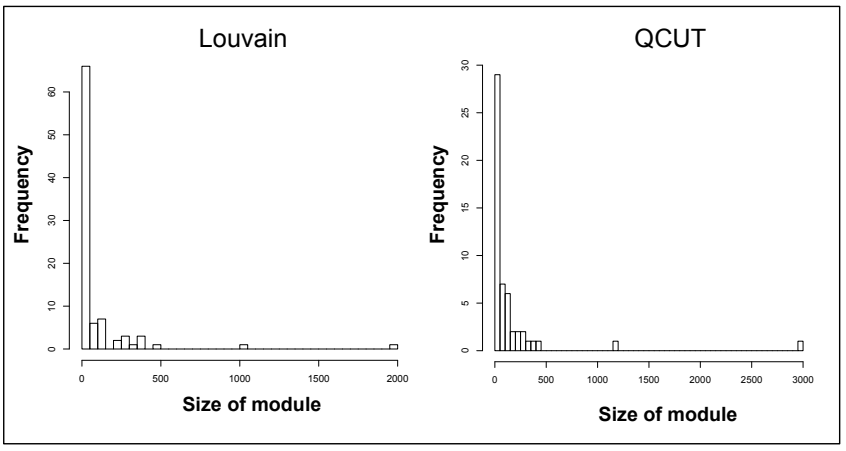

FIGURE 6.2: The module size distribution of the hard partitions of the *F. graminearum network detected by Louvain and QCUT.*

and returns a non-redundant set of MICA terms that are relevant to at least a certain fraction of the input genes. A trade-off is made between the level of coverage of a term and the level of specificity of that term. The AIC-MICA statistic, AIC, for a module is an average of the IC values of any associated terms. For more details, see Chapter 2, Section 2.2.4.1.

Annotations for all three aspects of GO (biological process (BP), molecular function (MF) and cellular compartment (CC)) were considered for the communities in the Louvain partition with at least 5 annotated nodes and the AIC-MICA approach was used to find the most specific terms applicable to at least 60% of the nodes. It is found that 43 communities are assigned a term from the BP aspect of the Gene Ontology, 52 are assigned a term from the MF aspect of GO and 35 are assigned a term from the CC aspect of GO. Figure 6.3 shows the corresponding MICA BP terms and their percentage of coverage for the largest communities. Some highly functionally coherent communities detected were "transport" and "carbohydrate metabolic process" (communities 3, 31 and 88 respectively, with 100% coverage) and "oxidation-reduction process", "transport" and "regulation of transcription, DNA-dependent" (communities 28, 60 and 76 respectively, with coverage >90%). Other communities with a strong functional coherence correspond to "vitamin transport" (community 78), "nucleotide biosynthetic process" and "serine family amino acid metabolic process". Expectedly, larger communities show less homogeneous functional content and therefore a broader GO term is assigned. For example community 79, the largest community is assigned the general term "cellular

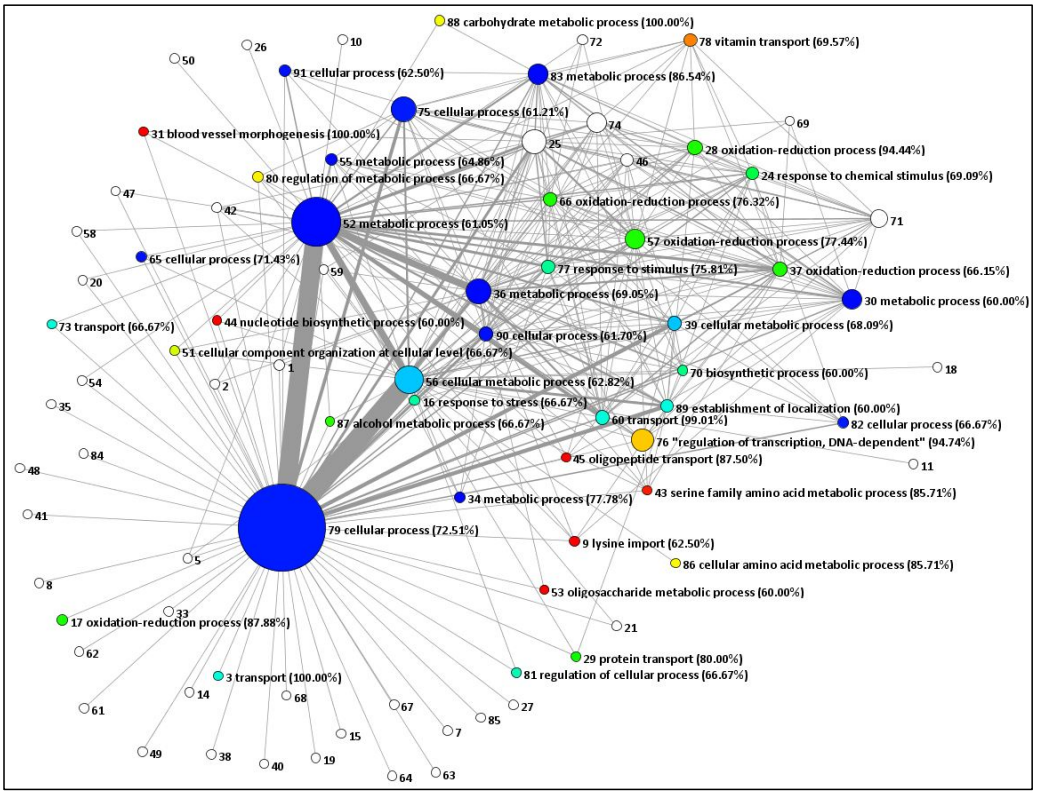

FIGURE 6.3: The meta-view of the hard partition of the main component detected by the Louvain method, where nodes represent communities. The thickness of the links between the communities corresponds to the number of genes that are shared between communities in the overlapping community structure discussed in Section 6.3.2. For the larger communities, the MICA (BP) term is shown next to the corresponding community and the corresponding percentage of coverage (visualisation generated in Ondex [104, 132]). Figure taken from [31].

process". Overall, the hard partition detected by the Louvain method appears to find some strongly biologically coherent communities.

6.3.2 Overlapping community structure

The hard partition of the main connected component is converted to a soft partition with overlapping communities using the mathematical programming method, OverMod, described in Chapter 5. The hard partition results in 3877 border nodes (nodes which interact with nodes in other modules), which are the potential multi-clustered nodes. The community membership of the remaining 4487 isolated nodes that are only associated

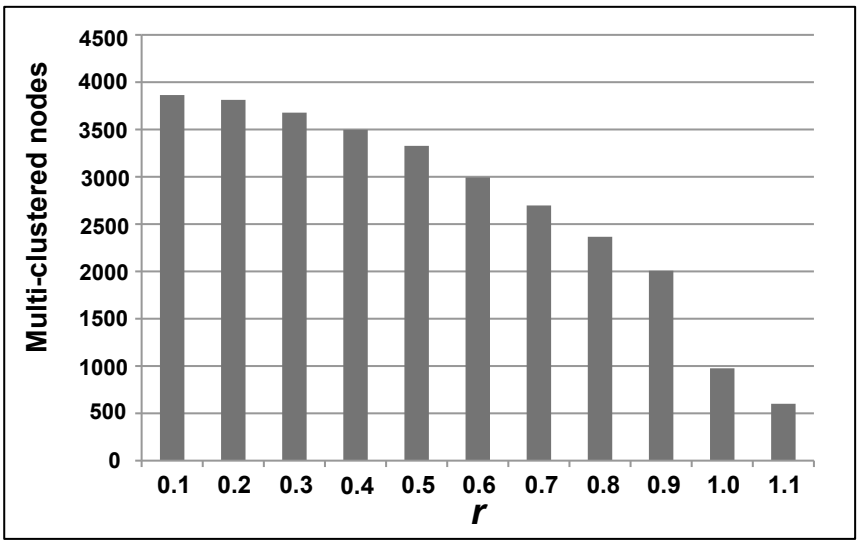

FIGURE 6.4: Multi-clustered nodes detected by OverMod for $0.1 \leq r \leq 1.1$.

to intra-community edges, are fixed and do not change in the course of the conversion procedure. In other words, the MINLP in OverMod is solved with only border nodes allowed to be assigned to multiple communities. Figure 6.4 shows the results for r ranging from 0.1 to 1.1 with $K = 1$ (the only value of K considered throughout this section). The suitability of this range is discussed in forthcoming sections. Table 6.2 shows how the number of communities that a multi-clustered node belongs to changes with r. It is found that at $r = 0.1$ the multi-clustered nodes belong to up to 13 communities, but as r increases, and the extent of overlap decreases, this range also decreases. At $r = 1.1$ multi-clustered nodes only belong to two communities maximum. Properties that characterise the nodes multi-clustered by OverMod are discussed in the following sections.

6.3.3 Evaluation of the multi-clustered genes

Here, like with the rat and human PPI networks in Chapter 5, the connectivity and multi-functionality of the multi-clustered proteins in the integrated *Fusarium* network are explored. These features are once again used as indicators of topological and functional relevance of the role of multi-clustered proteins in the entire network. However in this analysis, the number of protein domains is also considered. Furthermore, an appropriate range of values for the overlapping parameter, r, can be proposed based on the results.

r	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
0.1	2191	860	421	208	97	36	23	19	3	3	2	1
0.2	2193	886	392	206	67	36	19	12	2	0	0	0
0.3	2228	873	343	148	55	22	6	3	0	0	0	0
0.4	2297	804	263	96	29	0	0	0	0	0	0	0
0.5	2330	718	208	66	4	0	0	0	0	0	0	0
0.6	2354	512	106	22	0	0	0	0	0	0	0	0
0.7	2305	319	59	14	0	0	0	0	0	0	0	0
0.8	2135	217	14	0	0	0	0	0	0	0	0	0
0.9	1868	138	1	0	0	0	0	0	0	0	0	0
1	960	16	0	0	0	0	0	0	0	0	0	0
1.1	601	0	0	0	0	0	0	0	0	0	0	0

TABLE 6.2: Number of communities the multi-clustered nodes detected by OverMod belong to for $0.1 \leq r \leq 1.1$.

r	**Multi-clustered**	**Mono-clustered**	**p-value**
0.1	27.43	11.97	$<$ 2.2e-16
0.2	27.15	12.38	$<$ 2.2e-16
0.3	26.82	13.06	$<$ 2.2e-16
0.4	26.98	13.46	$<$ 2.2e-16
0.5	27.37	13.66	$<$ 2.2e-16
0.6	28.13	14.09	$<$ 2.2e-16
0.7	28.33	14.73	$<$ 2.2e-16
0.8	29.34	15.08	$<$ 2.2e-16
0.9	28.91	16.02	$<$ 2.2e-16
1	26.67	18.11	$<$ 2.2e-16
1.1	31.95	18.12	$<$ 2.2e-16

TABLE 6.3: Significance values of the difference between the average node degree of multi- and mono-clustered nodes detected by OverMod.

First the average degree of the nodes with multiple community membership is compared with the equivalent values for nodes that belong to only one community. For $0.1 \leq r \leq 1.1$, multi-clustered nodes have a higher average degree than the mono-clustered nodes (Table 6.3). The population means are determined to be statistically significantly different or not using the Mann-Whitney-Wilcoxon U test as implemented in the R statistical computing environment [172], where a p-value < 0.01 is significant. For all values of r tested, the average node degree of the multi-clustered nodes is significantly larger than the average degree of the mono-clustered nodes with all p-values $< 2^{-16}$ (Table 6.3). This result shows that, similarly to the rat and human PPI networks, multi-clustered genes detected by OverMod in the *Fusarium* network tend to have a higher number of interactions than those that belong to only one community.

r	All GO	MF	BP	CC
0.1	6.18E-06	9.73E-02	1.61E-05	4.04E-03
0.2	3.40E-05	2.94E-01	1.44E-05	8.89E-04
0.3	1.04E-04	3.03E-01	6.43E-06	1.06E-03
0.4	1.62E-04	2.31E-01	4.40E-06	1.06E-03
0.5	1.95E-03	1.23E-01	2.09E-05	7.37E-04
0.6	7.23E-03	5.35E-01	1.29E-03	5.33E-04
0.7	7.74E-04	9.28E-01	5.96E-04	9.05E-04
0.8	3.48E-04	3.95E-01	5.72E-05	1.84E-03
0.9	4.92E-03	4.94E-01	4.91E-07	1.03E-04

TABLE 6.4: The significance values of the difference between the average number of GO annotations for multi- and mono-clustered nodes.

As in Chapter 5, Gene Ontology annotations are again employed as a representation of functional importance, where one expects multi-clustered genes to be associated with a higher number of GO annotations than those belonging to only one community. The number of GO terms annotated to multi-clustered genes is compared to that of mono-clustered genes to determine which group appears to be more highly multi-functional. The annotations are taken from the MIPS *Fusarium* database [12] and were filtered to remove any redundant parent terms. The *Fusarium* genome has 4915 genes annotated with 13,883 non-redundant GO terms from all three aspects of GO (BP, MF and CC). As mentioned previously, the complete genome sequence comprises 13,718 protein coding genes and therefore only roughly a third of the genome is annotated. In the integrated network, 4311 proteins are annotated with at least one aspect and 4251 have no annotations. When broken down into the three aspects, there are more proteins unannotated than annotated in the network.

Each gene in the main component of the *Fusarium* network is mapped to its GO terms where possible. The average number of GO terms for all three GO categories combined (ALL GO) and for BP, MF and CC is calculated (excluding those with no GO terms). The results are shown in Table 6.4. For ALL GO, BP and CC the multi-clustered proteins have a statistically significantly higher average number of GO terms than the mono-clustered proteins, for $0.1 \leq r \leq 0.9$. For MF, the average number of GO terms for the multi-clustered proteins is not significantly higher than mono-clustered nodes for all values of r. The above analysis is possibly influenced by a lack of comprehensive annotations and therefore was carried out keeping in mind that the functional nature of much of the network is unknown. Such an observation leads the way for future work on function prediction through network analysis methods for example, as described in Chapter 2, Section 2.2.4.1.

r	**Multi-clustered**	**Mono-clustered**	**p-value**
0.1	1.4366	1.2746	1.73E-12
0.2	1.4402	1.2737	1.50E-13
0.3	1.4464	1.2742	5.66E-15
0.4	1.4492	1.2784	2.47E-13
0.5	1.4483	1.2847	2.95E-11
0.6	1.4599	1.2869	7.02E-12
0.7	1.4734	1.2883	6.15E-14
0.8	1.4788	1.2969	3.54E-10
0.9	1.4638	1.3123	3.67E-07
1.0	1.4261	1.3408	2.20E-03

TABLE 6.5: The significance value of the difference between the average number of protein domains for multi- and mono-clustered proteins.

In addition to the average number of GO terms, the average number of distinct protein domains is also considered. A protein domain is a section of a protein sequence that is functionally independent entity from the rest of the sequence. A domain may appear in several different proteins and may be recombined in different arrangements with other domains to create proteins with different biological functions. The number of domains that a protein contains can therefore be associated with its degree of multi-functionality. The protein domain information was also downloaded from the MIPS *Fusarium* database [12]. Genes with no Pfam annotation were removed and only distinct domain annotations for each gene were retained. It is found that the multi-clustered proteins have on average a higher number of distinct protein domains than mono-clustered proteins for $0.1 \leq r \leq 1.0$ (Table 6.5). These results are also represented in Figure 6.5, where it can be seen more clearly that the multi-clustered proteins have the highest average number of protein domains at $r = 0.8$, after which this begins to decrease. Indicating that despite the values remaining significantly higher for multi-clustered proteins at $r = 0.9$ and $r = 1.0$ than for mono-clustered proteins, this is potentially a cut off point for r. Overall, this result is a reflection of the multi-functionality of multi-clustered proteins that was already derived in the GO terms analysis above and contributes to the hypothesis that multi-clustered proteins are more functionally relevant than mono-clustered proteins.

As seen in Chapter 5, the above features can indicate a suitable range of values for r based on the assumptions that the multi-clustered nodes identified by OverMod should exhibit properties that give evidence of their topological and functional relevance. In terms of node degree, these results suggest that the full range of values for r is reasonable for this network. In terms of multi-functionality: (i) number of protein domains indicate 0.1 to 1.0 and (ii) GO terms (using the ALL GO count) indicate a range of 0.1 to 0.9.

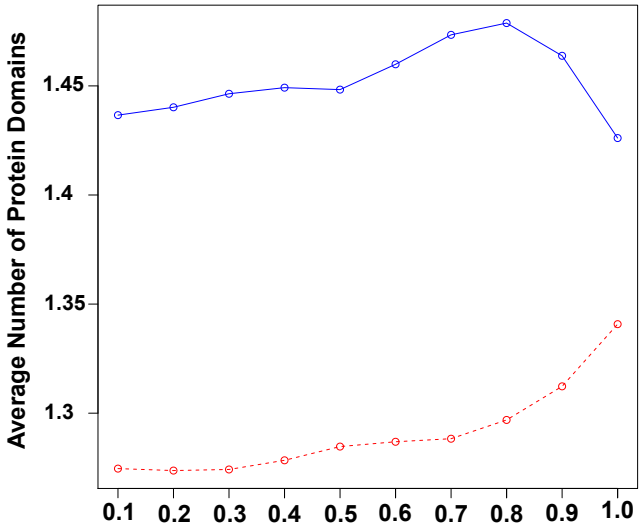

FIGURE 6.5: The plot shows the average number of protein domains of the multi-clustered (blue solid line) and the mono-clustered (red dashed line) proteins. It can be seen that the values for the multi-clustered proteins reaches a maximum at $r = 0.8$.

Taking all three measures into account, an appropriate range of values for r for the *Fusarium* network is $0.1 \leq r \leq 0.9$.

Overall, the results show that once again the combination of modularity optimisation method plus OverMod can successfully assign proteins to multiple modules with properties reflecting their important position in the system.

6.3.4 Functional cartography of multi-clustered genes

Throughout this study it is hypothesised that multi-clustered nodes play an important role topologically and functionally in the network, considering them as bridges or communicators between functional units, helping to maintain the structure of the system. This idea has been reinforced by showing that in the integrated network of *Fusarium*, multi-clustered proteins on average (i) are more connected than mono-clustered nodes, (ii) for some aspects of the Gene Ontology, have more functional annotations and (iii)

contain more protein domains than mono-clustered proteins. Here, the overlapping community structure of the network is related to a node role classification scheme proposed in [83] where each node is assigned a role based on its position in the hard partition of the network. A node's role is characterised according to two measures: within-community degree z-score and participation coefficient. The within-community degree z-score measures how well a node is connected with nodes in its own community and the participation coefficient measures how uniformly the nodes links are distributed among the other communities in the partition. More details can be found in Chapter 2, Section 2.2.4.2.

The node classification scheme can be summarised as follows. Based on the within-community degree z-score, nodes are classified as hubs and non-hubs, where hubs have a higher number of links with nodes in their own communities. Non-hubs are then classified into 4 roles: R1, ultra-peripheral nodes, R2, peripheral nodes, R3, non-hub connector nodes and R4, non-hub kinless nodes. Hubs are also classified into 3 roles: R5, provincial hubs, R6, connector hubs and R7, global kinless hubs. Both R3 and R6 nodes are labelled connector nodes according to the classification scheme as they have by definition a large participation coefficient, indicating a high distribution of links with communities other than their own. Consequently, the removal of these nodes may result in poorly connected communities or even the disconnection of communities and therefore impacting on the global structure of the network. On the application of the classification scheme to metabolic networks of 12 organisms it was found that R3 and R6 nodes are the most preserved across the species tested, suggesting that their role is more structurally relevant or in some cases essential. Similar results are predicted for other systems, including protein interaction and gene regulation networks [83].

Node roles are assigned to the *Fusarium* network based on the hard partition detected by Louvain in Section 6.3.1. The distribution of node role types is shown in Table 6.6. There are no R4 and R7 nodes, which is noted in [83] to be common. It is determined whether the proportion of R3 and R6 nodes is significantly higher in multi-clustered than mono-clustered nodes indicating that the multi-clustered nodes can indeed be described as connectors. For $0.1 \leq r \leq 0.4$, all 165 R3 nodes and all 50 R6 nodes belong to the set of multi-clustered nodes. For $0.1 \leq r \leq 1$, there are either all R3 nodes in the set of multi-clustered node or a higher proportion of R3 nodes in the multi-clustered set than the mono-clustered set. For R6 nodes, the range is $0.1 \leq r \leq 1.1$. It is found that, according to the Fisher's exact test, there is a significantly higher proportion of R3 and

Role types	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**
No. of nodes	4669	3323	165	0	157	50	0

TABLE 6.6: Node role type distribution.

r	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**	**1.1**
R3	8.45E-01	1.99E-01	2.76E-04	1.31E-07	1.61E-09	1.11E-07	1.08E-05	6.90E-05	3.96E-02	5.22E-01	1.38E-02
R6	8.45E-01	6.40E-01	7.05E-02	2.21E-02	1.19E-02	9.36E-05	1.08E-05	9.28E-07	3.74E-07	1.65E-01	1.41E-01

TABLE 6.7: The FDR-adjusted p-values for difference in proportion of R3 and R6 nodes in multi- and mono-clustered sets (significant values shown in bold).

R6 nodes in the multi-clustered nodes for $0.3 \leq r \leq 0.8$ and $0.6 \leq r \leq 0.9$ respectively (FDR-adjusted p-value < 0.01 is significant, Table 6.7).

The node classification scheme is employed to help better describe the topological nature of nodes that lie within the intersections of communities, offering a more sophisticated description than node degree alone. It is found that nodes described as connectors are significantly enriched in the multi-clustered nodes. These are node roles that have previously been shown to be more structurally relevant than other node roles in biological networks [83]. This goes some way to supporting claims that multi-clustered nodes play an important part in anchoring the communities of the network and thus contributing to the global functioning of the system. Furthermore, OverMod is successfully detecting such nodes.

Finally, R3 and R6 roles represent topologically very good examples of what is thought of as a multi-clustered node in this context. Therefore it is possible to implement the stricter criteria that R3 and R6 nodes should be enriched in the set of multi-clustered nodes detected by OverMod to suggest bounds on the range of appropriate values of r. Therefore, in the case of the *Fusarium* network, values of r from 0.6 to 0.8 inclusive would ensure the multi-clustered nodes were significantly enriched with connector nodes.

6.3.5 Verified virulence genes

The following analysis makes a preliminary attempt at linking the modular structure of the *Fusarium graminearum* network to known virulence genes. As described in [31], a set of 98 experimentally verified virulence (VV) genes known to be involved in different aspects of the infection and disease formation process of *Fusarium*, e.g. via gene

Community no.	7	16	28	39	51	52	56	57	64	71	75	76	79	80	82
No. of VVs	1	1	5	1	1	3	2	1	1	2	4	6	39	7	1
BP AIC	-	27.5	3.62	2.07	4.48	1.2	2.07	3.62	-	-	1.29	5.76	1.29	4.88	1.29

TABLE 6.8: Distribution of the verified virulence (VV) nodes among communities and corresponding biological process average information content (BP AIC).

knockout experiments, have either been extracted from the Pathogen-Host Interaction database (PHI-base) [212, 213] or manually obtained from the scientific literature. Here these virulence-associated genes are placed in the context of the community structure of the integrated network. Of the 98 VV genes, 79 are found to map to the integrated network, of which 75 are in the main component.

The distribution of the VV nodes in the disjoint communities detected by Louvain is first considered. These nodes appear in only 15 of the 91 communities (Table 6.8), with the largest community (community 79, Figure 6.3) containing over half of the VV nodes (39 out of 75). For each of the 15 communities, it is determined if the community has a statistically significant higher proportion of VV nodes than the rest of the network using Fishers exact test (an FDR-adjusted p-value < 0.01 is significant). Only the largest community and another of size 48 (community 80 in Figure 6.3), with 7 VV nodes, encompass a statistically significant high proportion of the proteins (FDR adjusted p-values 8.38E-07 and 1.35E-06 respectively). Although community 79 does contain a significant number of VV nodes, the corresponding BP MICA term, "cellular process", has an AIC of only 1.29 (Table 6.8), indicating that this is highly functionally diverse. Community 80 however is more coherent with an AIC of 4.88 (Table 6.8), and its BP MICA term is "regulation of metabolic process". The 7 VV genes in community 80 are all predicted to be transcription factors of the Zinc finger (Cys_2His_2) type [188]. Therefore module 80 not only has a relatively good degree of coherence in terms of its overall function (AIC equal to 4.88), but also in terms of its association with the infection process.

Due to the disproportionately large size of community 79 it is further partitioned in order to determine whether some underlying sub-community structure exists and the distribution of the VV nodes in this new hard partition is examined. The Louvain method detects a partition with 19 communities. The 39 VV nodes that are in community 79 in the hard partition of the original network are found in 8 of the communities of the re-partitioned community 79 (Table 6.9). Again, checking for overrepresentation of VV nodes in the individual communities shows that no community is significantly enriched.

Community number	1	5	7	9	14	15	16	17
No. of VV proteins	9	3	2	2	5	12	5	1

TABLE 6.9: Distribution of verified virulence (VV) proteins in the hard partition of community 79.

r	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1
No. of VV proteins	40	40	39	37	35	32	26	25	24	3	1

TABLE 6.10: The number of verified virulence (VV) genes that belong to more than one community for $0.1 \leq r \leq 1.1$.

This may be a result of the fact that proteins involved in virulence tend to belong to a wide range of processes, reflecting overall complexity of the infection process. This is demonstrated in PHI-base [212, 213], where many proteins are annotated with a "general" functional role such as metabolism, signal transduction and transcription and less with a specific function such as toxin biosynthesis or infection structure formation.

Next the connection between multiple community membership and the VV genes is considered. For $0.1 \leq r \leq 1.1$, the number of VV genes that are found by OverMod to belong to more than one community are shown in Table 6.10. The VV genes were not seen to be significantly overrepresented in either the multi-clustered genes or the mono-clustered genes for all values of r. Despite statistical tests showing no significant results here, the number of VV genes is relatively small and it is noted that roughly half of the VV genes are multi-clustered for $0.1 \leq r \leq 0.4$, an encouraging number that may increase in the future as more virulence-associated genes become known. Therefore a potential association of virulence genes to multiple modules should not be ruled out completely, a possibility that is further explored in the next section by considering genes that have been recently predicted to be related to virulence.

6.3.6 Predicted virulence genes

Considering the connection of a set of predicted virulence genes with the community structure of the *Fusarium* network further develops the result from the previous section. In a further study, potential virulence-associated genes were predicted using a guilt-by-association procedure [133]. Genes connected with at least two verified virulence (VV) genes in the integrated network were predicted to be involved in the *Fusarium* disease process, based on the assumption that genes important to the virulence procedure are more likely to be connected to other genes with similar properties [126]. This assumption

was verified in [133] through permuting node labels 10000 times in order to find an estimate of how likely it is that a gene annotated to be involved in virulence is connected with at least two others. It was observed that the probability is significantly higher than would be expected by chance. The prediction strategy generated 215 potential virulence genes that are connected to at least two and up to 8 VV genes.

It was found in the previous section that the VV genes did not show a significant tendency to belong to multiple communities although it was noted that for $0.1 \leq r \leq 0.4$ nearly half of the VV genes belonged to more than one community. However, the small number of VV genes makes it difficult to ascribe biological significance to these results. This previous result is now developed through the investigation of the topological nature of the 215 predicted virulence genes. It is found that 164 out of the 215 are in the largest module of the hard partition of the network (community 79 of size 1951 nodes), which also contains 33 VV genes. This module was shown to be significantly enriched for VV genes, and therefore, it makes sense that a large number of the predictions also belong to this community due the nature of the guilt-by-association procedure. The module memberships of the predictions are now explored to determine whether they have a tendency to belong to more than one module. According to the Fishers exact test, a statistically significant proportion (p-value < 0.01) of the prediction genes belong to more than one module for r in the range of 0.1 to 0.9 inclusive (p-values shown in Table 6.11). For r equal to 1 and 1.1, a larger proportion of the mono-clustered genes are predicted to be associated with virulence. This result may be due to the fact that multi-clustered genes tend to be more connected than genes belonging to only one module and therefore have a higher chance of being connected to the VV genes. However, it may also indicate that the genes predicted as being associated with virulence do indeed have a tendency to be multi-clustered.

Again, due to small number of VV genes, it is hard to make any solid conclusions about the nature of virulence-associated genes. However, this small development of the investigation from the previous section is already promising and one would expect that the inclusion of more experimental data would increase the utility of this approach.

6.4 Discussion and conclusions

In this chapter, the disjoint and overlapping community structure of an integrated network for the globally important plant pathogenic fungus, *Fusarium graminearum*, was

r	**Multi-clustered**	**p-value**
0.1	159	5.21E-17
0.2	157	1.32E-16
0.3	153	2.95E-16
0.4	145	1.43E-14
0.5	141	9.08E-15
0.6	127	1.73E-12
0.7	97	4.43E-05
0.8	90	1.07E-05
0.9	81	4.03E-06

TABLE 6.11: The number of predicted virulence genes that are multi-clustered and their significance scores.

explored. As a major threat to the yield of wheat and other cereal crops, research into the disease process of *Fusarium* is important due to the potential implications in the agricultural industry.

The integrated network of *Fusarium* brings together interactions from three data sources with the aim of building a more informative and reliable network than that of a single data source. The detection of the disjoint community structure revealed several coherent functional units according to the commonality of the GO annotations of member genes. Topological and functional properties of proteins belonging to more than one community were compared with those belonging to only one. Results show that proteins in the intersection between two or more communities tend to be more highly connected than mono-clustered proteins. Furthermore, a deeper level of topological description of the multi-clustered proteins was offered by the node role classification scheme in [83]. It was shown that the set of multi-clustered nodes are enriched with nodes with connector roles, with such nodes having previously been shown to be structurally relevant in biological networks in [83]. Additionally, the functional properties of the multi-clustered proteins in terms of number of GO annotations were explored. For all three aspects of GO combined (ALL GO) and for BP and CC multi-clustered proteins had a higher average number of annotations than proteins belonging to only one community, although the same trend was not seen for MF. The difference in the extent of multi-functionality was also reflected in the higher average number of domains contained in multi-clustered proteins. The GO annotation results for *Fusarium* are not as strong as for the rat and human PPI networks in Chapter 5. This may be in part due to the fact that *Fusarium* is a much less well-annotated organism. However, some significant results are still found and one would expect them to improve as more information about the pathogen becomes available. In the meantime, the fact that so much of the nature of *Fusarium* is unknown

makes it an ideal organism for exploratory analysis, where network tools such as those described in this study can be used to generate hypotheses concerning functional aspects of the organism.

Finally, the connection of the known virulence genes to the disjoint modular topology of the network was investigated. It was observed that the VV nodes were concentrated in two communities suggesting that the pathogenicity process is linked to particular pathways, rather than being distributed throughout the network. The relationship of the verified virulence genes to the overlapping community structure was also determined, although the VV genes were not shown to be significantly multi-clustered. However, in addition to the VV genes, the topological nature of 215 genes predicted to be associated with the disease process was also explored. It was found that the predictions did tend to belong to more than one module. Although this may be as a result of the means in which they were predicted, it may also indicate a tendency of virulence-associated genes to belong to more than one module.

Although there are no concrete conclusions regarding the virulence genes and their connection to the community structure of the network, this analysis can be seen as a first step in the direction of a more in-depth study, where candidate virulence genes can be predicted and verified in future experiments. It is also noted that within the set of known virulence-associated genes there may be a bias in reflecting particular classes of proteins that have been investigated experimentally, for example intracellular signalling and transcription-associated proteins. Therefore, network analyses and systems biology strategies such as the one presented in this chapter offer the possibility of planning future experiments using a more rational basis.

As mentioned above, the network analysed contains information from multiple heterogeneous data sources and some of the data sources may be of better quality than others. A way of improving the study would be to include weighted edges in the network. This might provide a more accurate and informative description of the community structure of the organism. A simple approach weights the edges heuristically depending on the number of data sources that suggest an association subject to the various thresholds chosen, this can be regarded as an indication of reliability of an interaction. As has been shown in Chapter 4, community structure detection of a weighted network may result in a different partition as compared to the equivalent binary network. Such effects can be addressed in future work.

The motivation behind this study was to gain a better understanding of the cellular organisation of the fungal pathogen, *Fusarium graminearum*. Network analysis tools were used to investigate the underlying mechanisms of the fungus from a community structure perspective. As the number of VV proteins increases, analytical methods featured in this study could prove promising in predicting more virulence-associated genes and gaining insights into infection-related pathways. Overall, this chapter has shown that the methodologies presented throughout this thesis thus far have the ability to extract meaningful biological results and therefore have the potential to play an important role in future bioinformatics applications.

Chapter 7

Community structure detection in dynamic networks

The network representations of complex systems that have so far featured in this thesis have been static. However, in reality many complex systems are constantly evolving. Consequently, static networks may not offer a sufficiently true to life abstraction of a complex system. Over the course of the previous chapters, the aim has been to develop more realistic modelling frameworks, first through the incorporation of weighted interactions, followed by allowing communities to overlap. The next challenge in community structure detection is therefore integrating network dynamics into clustering models.

In this chapter, consensus clustering of dynamic networks is tackled; defined as detecting a single partition of a dynamic network that is relevant across multiple snapshots. This is addressed by extending previous MIQP and MINLP models featured in Chapters 3 and 4 such that average modularity across network snapshots is now optimised. A comparison is made with a similar method from the literature on four dynamic networks showing that the proposed approaches achieve competitive results for small to medium sized networks, although scalability limitations are encountered. Overall, this chapter represents the next step in the search for a more true to life representation of community structure and identifies future work that will contribute towards its continuation.

7.1 Introduction

As discussed throughout this thesis, network analysis and related computational approaches, in particular community structure detection, have proven to be important tools in the investigation of the principles governing complex systems. Until now, such techniques have been discussed in the context of static networks. That is, networks that either represent a snapshot of a system at a certain point in time or an aggregation of data over multiple time points. However, in reality networks are not static; nodes and interactions can be created or equally they can cease to exist. For example, in social networks friendships are made and broken. In a business, employees retire and new members of staff are employed. Furthermore, in biological systems, not all interactions take place at the same time, depending upon spatial, temporal or environmental conditions [163]. The changes which occur at the node and interaction level are reflected at the global level, i.e. the community structure. Similar to nodes and interactions, modules can be created or destroyed and can even be split or merged together. Consequently, incorporating temporal information into network modelling frameworks may lead to a more accurate understanding of the underlying nature of a complex system.

It follows that the current challenge in community structure detection is the identification of modules in dynamic networks. A dynamic network is defined as a series of network snapshots at two or more time points where time can represent seconds, days, years or even, in a biological context, phylogenetic distance. The problem has already been studied in various ways, some of which will be discussed in the following section. Here, the aim of dynamic network clustering is to find a single partition of a dynamic network that is relevant at multiple time points, i.e. consensus clustering, with the reasoning that the most important or persistent modules will be uncovered.

This chapter unfolds as follows. First an overview of existing methods for community structure detection in dynamic networks is given. The consensus clustering problem is then tackled by proposing modifications to previous MIQP and MINLP formulations of modularity optimisation such that the objective function is now the average modularity across multiple time points. Both methods are compared with a similar algorithm from the literature on four test networks. Overall, this chapter lays the foundations of potential solutions to the problem of community detection in dynamic networks and outlines the next steps that will feature in future work.

7.2 Related work

The dynamic clustering problem has been approached in several different ways. Here a summary of a few existing methods is given.

First, the idea of detecting a single partition for a series of snapshots has been explored. In [25], Aynaud and Guillaume propose two methods for detecting a consensus partition for a dynamic network over a given period of time, where the partition is to some extent relevant at each time step. In the first method, an average representation of the network snapshots, the sum graph, is produced by combining all snapshots and weighting each edge by the length of time that it exists. The sum graph is then partitioned using the Louvain method [34] (see Chapter 2, Section 2.3.2), but can equally be partitioned using any other static network clustering algorithm. This first method is known as the sum-method. The second approach, although similar, but not equivalent, optimises the average-modularity across all snapshots of the network simultaneously by a modified version of the Louvain method. In the adapted version of the clustering algorithm, the average modularity gain is optimised where this is the average of the static modularity gains for each snapshot. In the original version of the Louvain method, once modularity can no longer be increased, a meta-network is constructed where the communities become the meta-nodes and the edges are weighted according to the number of links between communities. Here, in the dynamic version of the method, the same transformation from network to meta-network is applied to each snapshot in the dynamic network independently, thus creating a new dynamic network of meta-snapshots. Then, like with the original Louvain method, the above steps are repeated. This method is known as the "Average-method" and the consensus partition is known as a multi-step partition. It is shown by the authors that although considering one consensus partition for several snapshots would not be as relevant at specific time points as clustering each individual snapshot would be, the proposed methodologies can still find partitions with equal level of relevance over several time points.

Alternatively, many approaches cluster the static snapshot networks independently and employ various methods of comparison between partitions to quantify change in community structure or follow the evolution of communities. Palla et al. [158] use the clique percolation method to detect the overlapping community structure of each snapshot network. Communities are matched across time points by partitioning the network that is the union of two consecutive networks and measures are proposed for tracing the evolution of a community and for predicting its lifetime. Similar approaches are used in [66]

and [92]. Sun et al. [192] adopt the Minimum Description Length (MDL) principle to group similar network snapshots into time segments and detect change points representing drastic discontinuities in the network dynamics. Information theory principles are used to detect communities through minimising the encoding cost. Duan et al. group together similarly weighted directed network snapshots into time segments according to similarity between partitions [59]. Asur et al. use MCL to partition snapshots independently and, similar to the idea of change points in the MDL method, critical events in the evolution of a network are identified [24].

The ideas of consensus partitioning and comparing partitions across time points are combined by Lancichinetti and Fortunato [109]. It is first suggested as a method for tackling the instability of partitions in static networks found by the same method (see the discussion on solution degeneracy in Chapter 8, Section 8.4). However, it is shown that the procedure can equally be used to detect partitions in dynamic networks. The idea is that each snapshot is clustered and then a consensus partition is generated for the first r snapshots, i.e. from $t1$, $t2$, ..., tr and then from $t2$, ..., $tr + 1$ etc. (i.e. a moving time window) to form a series of consensus partitions for overlapping time points. The modules from each of the different consensus partitions are then related over time points using the Jaccard measure.

Where snapshots are clustered individually, historic community structure is not taken into account. It has been proposed, however, that the community structure of the network at time t should not be taken as independent of the community structure of the network at time $t - 1$. In other words, a network at time t is clustered with respect to a known partition of the network at $t - 1$. Such methods preserve previously acquired information about the community structure and in some cases enforce smoothness across partitions. This can also be thought of as updating or maintaining a partition, where changes to a network can be broken down into a series of simple events: addition or removal of a node or an edge. Not only does this reduce variation between partitions at different time steps, it also, in some cases avoids clustering from scratch at each time point.

Chakrabarti et al. [41] propose the first evolutionary clustering method and introduce the idea of temporal smoothness, where the snapshot quality measure (e.g. modularity) is maximised at the current time point and a distance measure, the history cost (e.g. mutual information, rand index etc.) between the current network and the network at the previous time point is minimised. In this way, a trade off is made between remaining

faithful to the current data, but minimising the variation between the current partition and the previous one. Görke et al. [77] maximise a combination of modularity and the rand index with some node-module allocations from the partition at $t-1$ fixed and the remaining nodes free to be reallocated, according to various prep strategies. They find that maintaining a dynamic clustering rather than recomputing from scratch results in higher modularities and due to the fixing of node-module allocations, reduces computational cost. Similar approaches can be seen in [46, 99, 123, 152, 151]. Chi et al. take a spectral approach to evolutionary clustering in [46]. In [123], Lin et al. solve the evolutionary clustering problem from a Bayesian perspective. In [152], Nguyen et al. first find a base partition of the network at $t0$ with the Louvain method [34] and the update the partition at each subsequent time point according the addition/removal of nodes/interactions. The same group similarly tackle overlapping community structure in dynamic networks in [151]. In [99], Kim and Han find smooth communities by modelling dynamic networks as a collection of particles called nano-communities and a community as a densely connected subset of particles called a quasi-l-clique-by-clique. Using a cost embedding technique, modularity is then optimised.

Community structure detection in dynamic networks has also been applied in a biological context. In biological networks, dynamics can represent evolutionary time or short time steps within a specific organism. In [67], it is shown that over evolutionary time, protein interaction networks become more modular by clustering a present day yeast protein interaction and ancestor networks obtained through orthologous categorisation of the yeast open reading frames (ORFs). Yeast is again explored in [196], where it is found that communities in networks generated from time course gene expression data are more biologically informative than communities detected from a static PPI network. These networks, known as TC-PINs (Time Course Protein Interaction Networks), are used in the method comparison in Section 7.3.3.3.

These two examples illustrate the potential applications of dynamic clustering of biological networks and emphasise the need for more accurate and robust methods to improve such analyses. This chapter proposes a solution procedure to tackle the consensus clustering approach to the dynamic clustering problem, as seen in [25] and [109], which is addressed in the following section.

7.3 Simultaneous clustering of multiple network snapshots

In previous chapters, mathematical programming models have been developed to detect communities in unweighted networks, weighted networks as well as to detect overlapping modules. Here, similar modelling frameworks are employed to detect a consensus partition of a dynamic network by optimising the average modularity across time points. The approach is illustrated in Figure 7.1. In [25], it is suggested that a consensus partition is relevant to some degree at each time point in the dynamic network and will represent the most presistent functional groups. Furthermore, in the case where a dynamic network is generated from noisy data, simultaneous clustering may compensate for effects produced by false positive and false negative interactions, assuming that the false positives and false negatives are indeed random. For example, microarray data is known to have a high rate of false positives and false negatives. It is possible that in time series gene expression data, two or more interactions which do occur simultaneously are detected at separate time points. Therefore, these associations would be missed if the gene expression networks generated for each time point were clustered individually. It is therefore proposed that simultaneous clustering may lead to uncovering functional groups that are important to the whole dyamic process being modelled and also associations between groups of nodes that may have been otherwise missed.

In this section, the problem of simultaneously clustering multiple network snapshots is formulated as two different mathematical programming models. The first method globally optimises the average modularity measure while the second finds locally optimal solutions. A comparative analysis is made with a similar method by Aynaud and Guillaume [25] on four dynamic network examples.

7.3.1 Exact simultaneous clustering: DynOptMod

Here, the MIQP modularity optimisation framework, OptMod [219] (described in Chapter 2, Section 2.3.6), is extended to optimise the average modularity over several network snapshots. The input is a series of undirected, binary network snapshots and the output is a consensus partition of the network. The new mathematical model, known as DynOptMod, and its associated indices and parameters are presented below:

Indices

n, e nodes (the union of all network snapshots)

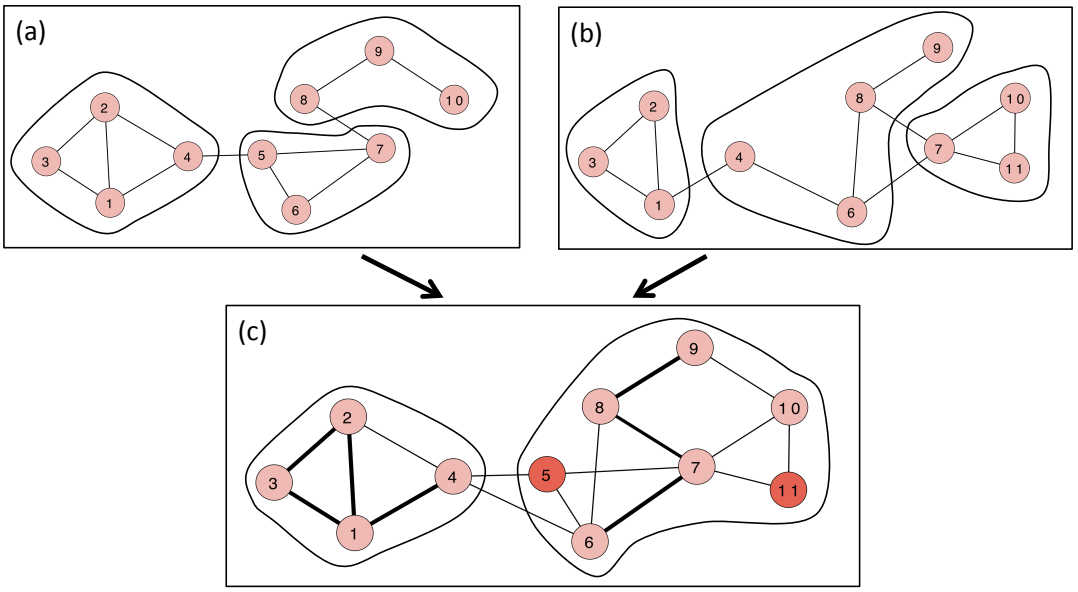

FIGURE 7.1: Simultaneous clustering: (a) Shows the partition of the snapshot of the network at time $t1$. The network is then modified; node 5 is removed and node 11 is added as well as various changes to edges. (b) shows the partition of the snapshot of the network at time $t2$. (c) shows the combination of both networks, where bolder edges are edges that appear at both time points and darker nodes are the nodes that do not appear at both time points. The partition in (c) is the consensus partition of both networks, that is, the partition representing the community structure over the two time points. The consensus partition differs from both individual snapshot partitions such that a compromise has been found between the two partitions.

l	links (the union of all network snapshots)
m, k	modules
t	time points

Parameters

β_{lt}	equal to 1 if the link l exists at time t
d_{nt}	degree of node n at time t
L_t	number of edges in the network at time t
T	total number of time points
M	total number of available modules

L_{total} total number of links over all time points

Sets

S M most connected nodes

AM_n allowable modules for node n

AV_m allowable nodes for module m

LM_l allowable modules for link l

B_n nodes with higher connectivity than node n

Continuous variables

L_{mt} number of links in module m at time t

D_{mt} degree of module m at time t

Binary variables

$$E_m = \begin{cases} 1 & \text{if module } m \text{ exists;} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{nm} = \begin{cases} 1 & \text{if node } n \text{ belongs to module } m; \\ 0 & \text{otherwise} \end{cases}$$

$$X_{lm} = \begin{cases} 1 & \text{if link } l \text{ belongs to module } m; \\ 0 & \text{otherwise} \end{cases}$$

The objective function is the average modularity across all snapshots in the dynamic network, defined as follows:

$$Q_{ave} = \frac{1}{T} \sum_t \sum_m \left[\frac{L_{mt}}{L_t} - \left(\frac{D_{mt}}{2L_t} \right)^2 \right] \tag{7.1}$$

Q_{ave} is maximised subject to several constraints which are described below.

First, L_{mt} and D_{mt} are defined as follows:

$$L_{mt} = \sum_{l} \beta_{lt} X_{lm} \quad \forall m, t \tag{7.2}$$

$$D_{mt} = \sum_{n} d_{nt} Y_{nm} \quad \forall m, t \tag{7.3}$$

where equation 7.2 is the sum of the weight of the edges in module m at time t and equation 7.3 is the sum of the degree of the nodes in module m at time t.

The clustering problem has returned to detecting a partition of disjoint communities, therefore the following constraint ensures each node belongs to exactly one module:

$$\sum_{m} Y_{nm} = 1 \quad \forall n \tag{7.4}$$

However, as with OptMod, nodes are allocated to certain modules according to their connectivity. In order to avoid equivalent solutions produced by re-numbering modules, nodes can only be assigned to a particular set of modules, AM_n. To do this, the nodes are first ordered according to their connectivity. The connectivity of a node in this case is calculated by summing the degree of a node over all time points. If $n1$ is the most connected, then it can only belong to module $m1$, if $n2$ is the second most connected then it can be assigned to modules $m1$ and $m2$ and similarly for the remaining $M-2$ most connected nodes in set S. The remaining nodes that are not in set S can be assigned to any of the M modules. Therefore constraint 7.4 becomes:

$$\sum_{m \in AM_n} Y_{nm} = 1 \quad \forall n \tag{7.5}$$

DynOptMod assigns a link l to a module m if both nodes associated with it, n and e, are also assigned to module m. In light of the above, this is only true if m is in both the sets AM_n and AM_e, resulting in the following three constraints:

$$X_{lm} \leq Y_{nm} \quad \forall m \in ML_l, l = \{n, e\} \tag{7.6}$$

$$X_{lm} \leq Y_{em} \quad \forall m \in ML_l, l = \{n, e\} \tag{7.7}$$

$$X_{lm} = 0 \quad \forall l, m \notin ML_l \tag{7.8}$$

where $ML_l = AM_n \cap AM_e$.

A number of additional constraints are included in the model and are formulated mathematically below. First, a degeneracy constraint is included to enforce that module m is only allowed to exist when the previous module, m-1 exists:

$$E_m \leq E_{m-1} \quad \forall m = 2, ..., M \tag{7.9}$$

i.e. if module $m-1$ is empty, module m is also empty. Module m is not empty if module m exists (i.e. $E_m = 1$) and the following two constraints are active at the same time:

$$\sum_l X_{lm} \leq L_{total} E_m \quad \forall m \tag{7.10}$$

$$\sum_l X_{lm} \geq E_m \quad \forall m \tag{7.11}$$

That is, module m contains at least 1 link if active (i.e. $E_m = 1$), and at most all links in the network (here we are not concerned with imposing bounds on the size of the modules, as described in Chapter 2, Section 2.3.6).

Furthermore, node n can only be allocated to module m (such that $m \in AM_n$) if at least one of the nodes with a higher connectivity than node n (members of the set B_n) which can be assigned to module $m-1$ (also a member of the set AV_{m-1}) has been assigned to module $m-1$. This can be formulated mathematically as:

$$Y_{nm} \leq \sum_{e \in (B_n \cap AV_{m-1})} Y_{em-1} \quad \forall n \geq 3, m = 3, ..., |AM_n| \tag{7.12}$$

In [219], it is claimed that computational cost can be reduced by only considering the M most connected nodes (i.e. set S):

$$Y_{nm} \leq \sum_{e \in (B_n \cap AV_{m-1})} Y_{em-1} \quad \forall n \in S, n \geq 3, m = 3, ..., |AM_n| \tag{7.13}$$

The resulting MIQP model, DynOptMod, comprises a concave quadratic objective function that is maximised subject to a set of linear constraints and mixed binary/continuous variables. DynOptMod is formulated as follows:
Maximise:

$$Q_{ave} = \frac{1}{T} \sum_t \sum_m \left[\frac{L_{mt}}{L_t} - \left(\frac{D_{mt}}{2L_t} \right)^2 \right] \tag{7.14}$$

subject to: Constraints (7.2, 7.3, 7.5-7.11 and 7.13)

$$L_{mt},\, D_{mt} \geq 0 \quad \forall m, t \tag{7.15}$$

$$E_m,\, X_{lm},\, Y_{nm} \in \{0, 1\} \quad \forall n, m, l \tag{7.16}$$

DynOptMod is implemented in GAMS [172] and the CPLEX mixed integer optimisation solver [13] (see Section 2.3.6 for a description) with 0% margin of optimality solves the proposed model to global optimality through the branch-and-bound procedure. Computational results are shown in Section 7.3.3.

7.3.2 Locally optimal simultaneous clustering: SimMod

The previous method, DynOptMod, optimises the average modularity to global optimality and therefore like OptMod, is likely to suffer from scalability limitations. With this in mind, the dynamic network clustering problem of detecting a consensus partition across multiple network snapshots is now formulated as an MINLP model.

The method described in Chapter 4, WeiMod, is now extended such that two or more networks are clustered simultaneously. As with DynOptMod, the input is a series of undirected network snapshots, the average modularity over all time points is optimised and the output is a single partition associated with each snapshot. Like with WeiMod, the input networks can have weighted interactions and nodes can have self-interactions (loops). The new model, known as SimMod, and its associated indices and parameters are presented below:

Indices

n, e	nodes (the union of all network snapshots)
m	modules
t	time

Parameters

β_{net}	weight of the link between nodes n and e at time t
d_{nt}	strength of node n at time t

α_{nt}	the weight of the loop of node n at time t
L_t	sum of the weights of all edges in the network at time t
T	total number of time points

Continuous variables

L_{mt}	the sum of the weights of the links that are in module m at time t
D_{mt}	the sum of the weighted degrees of nodes in module m at time t

Binary variables

$$Y_{nm} = \begin{cases} 1 & \text{if node } n \text{ belongs to module } m; \\ 0 & \text{otherwise} \end{cases}$$

The objective function is the average modularity over all time steps (as defined for DynOptMod in equation 7.14) and is optimised subject to the constraints described below.

First, all modules are disjoint, and therefore each node in the network can only be allocated to one module:

$$\sum_m Y_{nm} = 1 \quad \forall n \tag{7.17}$$

The objective function involves two continuous variables, D_{mt} and L_{mt}. The total degree of module m at time t, D_{mt}, is calculated by:

$$D_{mt} = \sum_n d_{nt} Y_{nm} \quad \forall m, t \tag{7.18}$$

where the weighted degree of a node n is defined as $d_{nt} = 2\alpha_{nt} + \sum_e \beta_{net}$.

Finally, a link is in module m at time t if both of its associated nodes, n and e, are also in m. Therefore, the total sum of the weights of all links in module m at time t, L_{mt}, is defined bt the following nonlinear equality:

$$L_{mt} = \sum_n \alpha_{nt} Y_{nm} + \sum_{\substack{n,e \\ n>e}} \beta_{net} Y_{nm} Y_{em} \quad \forall m, t \tag{7.19}$$

where node e is connected to node n at time t.

The resulting MINLP model (SimMod) comprises a non-linear objective function with a combination of integer and continuous variables and is formulated as follows:
Maximise:

$$Q_{ave} = \frac{1}{T}\sum_{t}\sum_{m}\left[\frac{L_{mt}}{L_t} - \left(\frac{D_{mt}}{2L_t}\right)^2\right] \tag{7.20}$$

subject to: Constraints (7.17-7.19)

$$L_{mt},\, D_{mt} \geq 0 \quad \forall m, t \tag{7.21}$$

$$Y_{nm} \in \{0, 1\} \quad \forall n, m \tag{7.22}$$

SimMod is a non-convex model and as such finds locally optimal solutions. Therefore, the MINLP is solved iteratively 100 times, each time with a different random initial solution to give a good representation of solution space. The best partition is taken as the solution with the largest value of Q_{ave}. SimMod is implemented in GAMS (General Algebraic Modelling System) [172] with the SBB mixed integer optimisation solver [1] (see Section 4.4.2.3 for a description) and CONOPT as the default NLP solver while relative and absolute gaps set to zero. Computational results are presented in Section 7.3.3

7.3.3 Computational results of DynOptMod and SimMod

In this section, a comparative analysis between DynOptMod, SimMod and the Average-method of Aynaud and Guillaume [25] is presented. The Average-method also optimises average modularity via a greedy optimisation method based on the Louvain method. At this point the evaluation criteria is based uniqely on the value of average modularity detected by each method. An implementation of the Average-method was downloaded from [82]. All experiments were run remotely on a bioinformatics Sun Fire X4450 Server running 16 Xeon(R) E7340 processors at 2.4GHz and 32GB of PC2-5300 667 MHz ECC fully buffered DDR2 memory. The server runs CentOS Linux release 5.8 OS. Results for four dynamic networks of varying size and nature are presented below.

Time	**Nodes**	**Links**	**Density**	***Q* (OptMod)**	**Modules**
***t*1**	34	78	0.1390	0.4198	4
***t*2**	33	75	0.1420	0.4066	4
***t*3**	32	63	0.1270	0.4015	4
***t*4**	31	62	0.1333	0.4010	4
***t*5**	30	59	0.1356	0.3885	4
***t*6**	29	57	0.1404	0.3744	4
***t*7**	28	51	0.1349	0.3258	3
***t*8**	27	50	0.1425	0.3214	3
***t*9**	20	30	0.1579	0.2733	3
***t*10**	19	29	0.1696	0.2646	3
***t*11**	18	27	0.1765	0.2407	2
***t*12**	17	25	0.1838	0.2200	2
***t*13**	16	24	0.200	0.2075	2
***t*14**	15	21	0.200	0.2211	2
***t*15**	14	19	0.2088	0.2368	2
***t*16**	13	16	0.2051	0.2500	2
***t*17**	9	9	0.2500	0.2716	2

TABLE 7.1: Summary of the dynamic karate network snapshots, with optimal partition found by OptMod [219].

7.3.3.1 The dynamic karate network

The Average-method software includes a test dynamic network with 17 snapshots, where the network at time $t1$ is the karate network [224] and time points $t2$ to $t17$ are modified versions of the original network. Each subsequent network is a subset of nodes and links from the previous network. The snapshot networks are described in Table 7.1, with number of nodes, links, snapshot density and optimal modularity as detected by OptMod [219].

Tests were carried out on (i) the first two snapshots, (ii) the first 3 snapshots, (iii) the first 10 snapshots and finally (iv) all 17 snapshots of the dynamic karate network. The results for DynOptMod, SimMod and Average-method are shown in Table 7.2. The values of average modularity found by DynOptMod are the global optimum values, therefore it can be seen that SimMod also finds the global optimum for the four tests. The Average-method finds suboptimal results for tests (i) and (ii) but globally optimal solutions for tests (iii) and (iv). Table 7.2 also shows CPU time for each experiment for DynOptMod and SimMod. Unsurprisingly DynOptMod is significantly more computationally expensive than SimMod, especially in tests (iii) and (iv). For this small

	DynOptMod			**SimMod**			**Average-method**	
	Q_{ave}	**Modules**	**CPU**	Q_{ave}	**Modules**	**CPU**	Q_{ave}	**Modules**
$t1-t2$	0.4120	4	3.00	0.4120	4	0.33	0.4090	4
$t1-t3$	0.4057	4	3.73	0.4057	4	0.60	0.4026	4
$t1-t10$	0.3698	4	121.47	0.3698	4	2.06	0.3698	4
$t1-t17$	0.3190	4	182.07	0.3190	4	3.97	0.3190	4

TABLE 7.2: Comparison of DynOptMod, SimMod and Average-method on the dynamic karate test network.

example this is not a problem however, as with OptMod it is expected that DynOptMod will show scalability limitations as network size or snapshot number increases.

The small example of the karate dynamic network has been constructed for the purpose of testing these methods but is an unrealistic example, therefore now three real life dynamic networks are considered.

7.3.3.2 The Enron email dynamic network

The Enron email dynamic network is commonly used as a benchmarking network for dynamic clustering method development. A brief summary of the background of the data is given below.

In just 15 years the Enron Corporation, an American energy, commodities, and services company, became the seventh largest business organisation in the USA, with 21,000 employees in over 40 countries. However, in 2001, Enron filed for bankruptcy and found itself under investigation by the US Securities and Exchange Commission (SEederal Energy Regulatory Commission (FERC) over illegal accounting and business practicC) and the Fes. For a detailed description of the Enron story, see [56]. Subsequently, a corpus of emails between around 150 employees including CEOs Jeffrey Skilling and Kenneth Lay, was publicly released by the FERC. The Enron email dataset comprises emails from a core set of 151 employees distributed in over 3000 folders. These folders contain emails to/from employees outwith the core set as many emails have more than one recipient, but since the complete email information of only 151 employees is available, only emails that were sent to/from core employees are used in constructing a dynamic network of email communication in the company.

Tests on DynOptMod, SimMod and the Average-method were carried out using a cleaned up version of the dataset by Shetty and Adibi [184]. An email communication network

was constructed for each month from January 2001 up to and including January 2002. During this period, the following major events occurred [56]:

- August 2001: Skilling resigned as CEO and former CEO Lay takes the position.
- October 2001: The Enron crisis fully breaks out and the stock market responds with an immediate drop and incessant decrease in Enron's shares.
- November 2001: Investigation fully underway.
- December 2001: Enron files for Bankruptcy.
- January 2002: Lay resigns as CEO.

Weighted snapshot networks were constructed from the dataset where the weight of an interaction between two employees is the number of emails they exchanged in that month. The network snapshots are described in Table 7.3. In each of the following network clustering experiments only the main component of each snapshot was considered. Note that the network snapshot at 2001-10 is significantly more dense than any of the other months, this corresponds to the time when the full Enron crisis broke out in full and stock market shares begin to rapidly drop.

For a more accurate analysis, weighted interactions should be taken into consideration. For example, the densities of the unweighted snapshots are more or less the same for each month, but there is much more deviation for the weighted networks (Table 7.3) and therefore the presence of weights or not should affect the final results. However, the downloaded implementation of Average-method is applicable to unweighted networks only and therefore in order to make a fair comparison, the networks are treated as binary, where a link exists when at least one email has been exchanged between the two employees in the corresponding month. As a preliminary test, the first three snapshot networks are clustered simultaneously (2001-01, 2001-02 and 2001-03), followed by clustering all snapshots. The results can be seen in Table 7.4.

First, as predicted from the high CPU time on the small dynamic karate network, clustering all 13 network snapshots proves to be too computationally expensive for DynOptMod and the experiment is terminated by the solver as resource limits are reached (100000 seconds). Therefore it is concluded that DynOptMod is really only useful in the case of benchmarking the locally optimal methods on smaller networks to gauge how well they perform. For any larger networks, DynOptMod will not be included in the comparison.

Snapshot	Nodes	Emails exchanged	Connected Components	Nodes in main	Emails in main	Density of main
2001-01	95	828	4	85	815	0.2283
2001-02	92	1227	1	92	1227	0.2931
2001-03	94	1419	2	92	1418	0.3387
2001-04	107	1396	2	105	1395	0.2555
2001-05	123	1464	2	121	1463	0.2015
2001-06	120	736	1	120	736	0.1031
2001-07	107	1089	1	107	1089	0.1920
2001-08	130	1776	1	130	1776	0.2118
2001-09	128	2544	1	128	2544	0.3130
2001-10	133	7287	1	133	7287	0.8301
2001-11	127	5105	1	127	5105	0.2631
2001-12	113	1688	1	113	1688	0.2668
2002-01	111	2265	2	109	2264	0.3846

TABLE 7.3: Monthly network snapshots of email interactions between the core set of 151 Enron employees.

	2001-01 to 2001-03			**2001-01 to 2002-01**		
Method	Q_{ave}	**CPU**	**Modules**	Q_{ave}	**CPU**	**Modules**
DynOptMod	0.6185	2.30e04	5	-	-	-
SimMod	0.6185	3.31	5	0.6175	60.05	8
Average-method	0.6157	-	5	0.5481	-	7

TABLE 7.4: Results of SimMod, DynOptMod and Average-method for the unweighted Enron networks.

Second, it is found that in both cases SimMod achieves marginally higher average modularity than the Average-method. Furthermore, for the first three snapshots SimMod achieves the global optimum where the Average-method does not.

7.3.3.3 Application to biological dynamic networks

In this section, SimMod and Average-method are applied to two biological dynamic networks that have been constructed from gene expression data. At this point in the method development, concern lies only with the method's ability to maximise average modularity and therefore evaluation of the biological significance of the consensus partitions will be considered at a later time.

The first dynamic network comprises 6 snapshot networks generated from gene co-expression data obtained from the ArrayExpress database [160] (Accession Number:

E-GEOD-76). The dataset consists of time course experiments monitoring 18 mice undergoing surgical intervention for pressure-overload induced hypertrophy and 18 sham-operated controls. Measurements were taken at 1 hour, 4 hours, 24 hours, 48 hours, 1 week, and 8 weeks post surgery. The raw expression values were normalised and filtered and the correlation of gene expression between samples were calculated and outliers removed. Overall, 4348 genes across 35 samples were found to be differentially expressed and considered for network inference. Pairwise gene co-expression matrices were constructed for each time point. Pairwise similarity in gene expression vectors was calculated by the Pearson correlation coefficient. Gene pairs that were correlated above a certain threshold were used to construct an undirected unweighted network, where nodes correspond to genes and links represent co-expression between genes.

The resulting 6 network snapshots are clustered by both SimMod and Average-method. SimMod finds a consensus partition of 5 modules with $Q_{ave} = 0.2448$ and Average-method finds a partition of 6 modules with $Q_{ave} = 0.1897$. Again, SimMod shows an improvement in results compared to Average-method. However, it is noted that in the case of both networks, the value of average modularity is relatively low; it is generally considered that a value of modularity equal to 0.3 indicates the presence of community structure [49], therefore a similar value for average modularity could also be assumed. This may be due to the fact that the network at 1 week is very dense with more than half the possible interactions existing (0.5879) and therefore may not be very modular and as such effects the whole consensus partition. In any case, here the methods are being evaluated with regard to value of average modularity, and in this case SimMod outperforms Average-method.

The biological meaning of the results was briefly investigated by looking at the functional enrichment of GO terms in each of the consensus modules. Initial results did not show the consensus modules to be significantly enriched for any functions. There are a few possible explanations, including the fact that there is very little overlap between these network snapshots, i.e. many nodes and links do not appear at multiple time points, and therefore this data may not be suitable for this type of analysis but may be more suited to evolutionary clustering where focus is on tracking the changes in community structure over time rather than the commonality. Furthermore, the problem could lie with the data and the fact that with gene expression data, there may be missing or false information. Equally it may be necessary to re-think the way in which the networks are constructed. However, only tentative steps were taken to biologically evaluate these results due to time constraints, and therefore currently, other than assessing the method's performance

Snapshot	Nodes	Links	Density
1 hour	483	5250	0.0451
4 hours	361	12527	0.1928
24 hours	590	18583	0.1069
48 hours	633	35998	0.1800
1 week	45	582	0.5879
8 weeks	200	5129	0.2577

TABLE 7.5: Summary of the time course gene co-expression network snapshots.

Snapshot	Nodes	Links	Density
TC-PIN 1	3450	14413	0.0024
TC-PIN 2	3404	14038	0.0024
TC-PIN 3	3460	15082	0.0025

TABLE 7.6: Summary of the first three TC-PINs.

based on value of average modularity obtained, no conclusions are made regarding the utility of the method for biological applications based on these findings.

Finally time course yeast protein interaction networks (TC-PINs) as described in [196] are considered. A static PPI network of *S. cerevisiae* was downloaded from the Databse of interacting proteins (DIP) [217] comprising 4,950 proteins and 21,788 links. Microarray data with gene with expression measured at 36 time points was obtained from Gene Expression Omnibus [29] (Accession Number GSE3431). Networks, known as Time Course Protein Interaction Networks (TC-PINs), were constructed for each time point by creating interactions if two interacting proteins in the static PPI network were also expressed above a certain level at that particular time point.

These network snapshots are significantly larger than previously tested networks and therefore only the first three TC-PINs are clustered in order to determine whether the methods are applicable to such large networks. The three network snapshots are summarised in Table 7.6. SimMod finds a consensus partition of 25 modules with $Q_{ave} = 0.4995$ and Average-method finds a partition of 24 modules with $Q_{ave} = 0.5189$. Therefore in this case, Average-method outperforms SimMod, which demonstrates the MINLP method's limitations in terms of scalability. In Chapter 4, WeiMod also shows signs of limitations in the case of the email network. Therefore scalability clearly presents a challenge for the mathematical programming models presented in this thesis. This will be discussed further in Chapter 8, Section 8.5.

It is again emphasised that this section serves only to evaluate the applicability of SimMod in comparison with Average-method and further analysis into whether the consensus partitions are biologically meaningful in these specific experiments and biological networks in general, will feature in future work.

7.4 Discussion and conclusions

In this chapter, two mathematical programming models were proposed to detect consensus partitions of dynamic complex networks. DynOptMod and SimMod, extended previous algorithms to simultaneously cluster a series of dynamic network snapshots by optimising average modularity across all time points.

A comparative analysis between DynOptMod, SimMod and a similar method from the literature, Average-method [25], was carried out. The simultaneous clustering methods were evaluated in terms of their ability to optimise Q_{ave}. First, it was found that DynOptMod, like OptMod, was only applicable to smaller networks due to the high computational cost of the MIQP model. Second, improvements were shown by SimMod over the Average-method on small to medium sized dynamic networks. However, for the largest dynamic network, the TC-PINs, Average-method was shown to outperform SimMod. Since the Average-method is a modified version of Louvain [34], which is known to be fast and accurate, it is unsurprising that Average-method also performs well on large dynamic networks. Overall, SimMod was shown to be a competitive method for clustering medium sized dynamic networks.

It is noted that there is a large difference in size between the second largest network (generated from mouse gene expression data) and the TC-PINs. Further tests on intermediate sized dynamic networks are required to determine more precisely the limitations of SimMod. In general, scalability limitations of the mathematical programming models presented in this thesis is a recurring issue, as it is with many modularity optimisation methods. The added complication in the case of dynamic networks is that not only the number of nodes, but also the number of snapshots have to be considered. Means of improving scalability will feature in future work and are discussed further in Chapter 8, Section 8.5.

In parallel to assessing the capabilities of DynOptMod and SimMod as optimisation models, the meaningfulness of simulataneous clustering as an approach to dynamic network analysis must also be considered; what does the consensus partition signify for a dynamic network? In [25], optimising the average modularity is said to uncover modules that are to some extent relevant at each time point and that correspond to the most persistant or prominent modules over the observed time period. For example, in the case of the Enron dynamic network the modules of the consensus partition may represent the groups of employees that remained faithful to one another during the crisis. As such, these modules may uncover information about which employees were in league with each other and therefore assist with any investigation. The meaningfulness of the results of the biological networks can be investigated by considering functional annotations. For example, the modules of the concensus partition may be enriched for certain biological pathways or functions. If so, it may be that these represent the most relevant biological functions in the process being modelled by the dynamic network. Overall, the results from simultaneous clustering may offer a more global view of the dynamic network.

However, major events or changes such as those outlined for the Enron network in Section 7.3.3.2 cannot be detected by a consensus partition. Equally, simultaneous clustering cannot track the evolution of modules. For example, in biological networks, it is of interest whether the modules enriched for certain biological functions remain constant throughout time, or whether the main biological functions change during the observation period. Such questions can be better addressed by evolutionary clustering.

As described in Section 7.2, evolutionary clustering is the partitioning of a network at the current time point while taking into consideration partitions from one or more previous time points. Future work could be envisaged towards the implementation of such an approach using a mathematical programming framework.

Clearly, dynamic networks can be approached in different ways depending on what type of questions are being asked about the system. It is therefore necessary to first define the problem statement of a specific experiment in order to decide which route to take. In summary, the steps required to advance the work presented in this chapter include applications to evaluate the results by SimMod, improvements to the method's scalability and finally implement a mathematical programming model to tackle the evolutionary clustering problem. Overall the methods presented in this chapter represent the first steps in addressing the concept of community structure in the context of dynamic networks and leave a clear pathway of further investigation.

Chapter 8

Conclusions

This thesis has investigated the problem of community structure detection in complex networks and in particular its relevance within the context of biological systems. A series of mathematical programming models have been developed to address various manifestations of the community structure detection problem. First the detection of disjoint communities in weighted and binary networks was tackled. This work was then extended to allow for the detection of overlapping modules. Finally, the concept of community structure in dynamic networks was explored. Where applicable, comparative analyses were carried out with methods from the literature. Results showed the proposed algorithms to be competitive with the associated approaches. Furthermore biological evaluations were undertaken to demonstrate the ability of the methodology to extract meaningful results in biological applications.

This chapter concludes the thesis. First, a brief overview of each chapter is given. Next, the research aims outlined in Chapter 1 are revisted in order to indicate where they have been addressed in the thesis and to assertain to what degree they have been fulfilled. Major contributions of the thesis are then detailed, followed by discussions regarding the limitations of the work and the potential avenues of future research. Finally, some concluding remarks are provided.

8.1 Overview of thesis

Chapter 1 introduced the general topic of research undertaken in this thesis; briefly explaining the rationale behind the work and placing the community structure detection problem in a bioinformatics context. This was followed by the outlining of the key research goals.

Chapter 2 provided a detailed review of essential background and related work. First an introduction to complex networks and their properties was given, followed by a description of several types of biological systems that can be modelled by complex networks. Community structure was then further discussed and in particular its significance to biological networks and example applications were described. Finally, the modularity measure was introduced and a review of modularity optimisation methods was given.

Chapter 3 tackled the problem of detecting a partition of disjoint communities in unweighted networks. An existing MIQP modularity optimisation model was incorporated into a novel two-stage mathematical programming approach known as iMod. An initial partition was first detected via an MINLP modularity optimisation formulation which was then improved through the iterative application of the MIQP algorithm. A comparative analysis showed that despite no guarantee of optimality, iMod achieved globally optimal solutions on networks with up to 512 nodes and furthermore outperformed all other methods tested when applied to larger networks up to 1133 nodes.

Chapter 4 saw the generalisation to weighted networks of the MINLP modularity optimisation model in Stage 1 of the iMod procedure, creating a clustering method known as WeiMod. Iteratively solving the MINLP 1000 times for each clustering experiment showed WeiMod to perform as well as, but in most cases better than three other modularity optimisation methods from the literature on weighted and unweighted networks with up to 889 nodes. However it was found that iMod performed better than WeiMod on unweighted networks, and in particular on a large network with 1133 nodes. The main conclusion is that WeiMod is a competitive method for clustering medium sized weighted and unweighted networks (defined loosely here as having < 900 nodes), however its scalability remains a point for consideration in future work.

Chapter 5 tackled the problem of detecting overlapping communities in complex networks. A two-stage procedure was proposed which first involved the detction of a hard

partition followed by the transformation of the disjoint communities to overlapping communities through an MINLP model. A method comparison for the karate network illustrated that results are highly variable across methods. Evaluations on two PPI networks found that the two-stage procedure assigned highly connected and multi-functional proteins to multiple modules. Moreover, many of the most strongly multi-clustered proteins were associated with disease, including cancer. Results suggest that the proposed methodology has the potential to detect structually and functionally important nodes in biological networks.

Chapter 6 investigated the disjoint and overlapping community structure of an integrated network of the major fungal pathogen *Fusarium graminearum*. The functional coherence of communities, properties of multi-clustered genes and the relationship of verified and predicted virulence-associated genes with community structure were all examined. This study was a first step in deciphering the underlying mechanisms of the pathogen. The analysis served to highlight the potential contributions of the methodology described in this thesis in future bioinformatics applications.

Chapter 7 addressed the final stage of the method development in this thesis by investigating the application of community structure detection to dynamic networks. A consensus clustering approach was taken to tackle the problem through the extension of previous MIQP and MINLP models to optimise the average modularity across network snapshots. Results showed the proposed methods to achieve competitive results for small to medium sized networks, although scalability limitations were encountered. However, evaluations of the significance of consensus partitions in a biological context are now required.

8.2 Research aims revisited

In Chapter 2, five research aims were outlined. These are now revisited in this section in order to ascertain how effectively they have been fulfilled.

- *To build on existing modularity optimisation methodology to detect disjoint communities in larger scale as well as weighted networks.*

  This goal was addressed in Chapters 3 and 4. The first step was tackled in Chapter 3 where the aim was to increase the scalability of the globally optimal MIQP

clustering algorithm, OptMod [219]. Although this required sacrificing the guarantee of optimality, the resulting method, iMod, still managed to retain OptMod's accuracy to some extent. iMod was shown to obtain globally optimal solutions for networks up to 512, an increase in scalability from networks of up to 104 nodes (OptMod alone). Furthermore, on networks with up to 1133 nodes, iMod found better sub-optimal results than four other modularity optimisation methods. Therefore, the first part of this research aim was achieved.

Weighted networks were addressed in Chapter 4. The decision was made not to include the MIQP component of the iMod procedure and to explore the capabilities of the MINLP formulation of modularity optimisation in its own right. WeiMod was shown to outperform methods from the literature on weighted networks with up to 889 nodes. However iMod was found to achieve better values of modularity than WeiMod on certain unweighted networks, most importantly on a network of 1133 nodes. Therefore in terms of clustering medium sized weighted networks (< 900 nodes), WeiMod has met the second part of this research aim. However the fact that iMod offers more accurate solutions on unweighted networks means that clustering larger weighted networks will feature in future work.

- *To define and implement a solution procedure to the problem of detecting overlapping community structure.*

  This aim was tackled in Chapter 5. The main challenge was to first define the problem statement. The route taken was a two-stage procedure which capitalised on previous clustering algorithms to first detect a disjoint partition which could then be transformed into a cover of overlapping communities via an MINLP model, known as OverMod. Evaluations of this method were not straight forward due to the variability of results shown across methods. However, applications to biological networks demonstrated the ability of the procedure to assign proteins to multiple modules which exhibited properties compatible with the general concept of a multi-clustered protein in the context of this thesis. It is therefore concluded that this research aim was met although further applications in biological networks will ascertain more robustly the capabilities of this methodology.

- *To define and implement a solution procedure to the dynamic community structure detection problem.*

This research aim was investigated in Chapter 7. Again, as with the overlapping community detection task, the iniital challenge was to determine which definition of the problem to explore. The consensus clustering approach was taken and MIQP (DynOptMod) and MINLP (SimMod) models were formulated. DynOptMod showed scalability limitations, which were expected since it guarantees global optimality. SimMod was shown to be a competitive method for clustering medium sized dynamic networks however also exhibited scalability limitations. Overall, Chapter 7 presented the first steps into clustering dynamic networks. Applications are required to verify the meaningfulness of the consensus partition in a biological context and therefore this research goal is still a matter of on-going work.

- *To evaluate methodology to show comparability with existing methods from the literature.*

  This research aim is covered in all chapters apart from Chapter 6. This aim refers generally to comparing the methodologies in this thesis against other methods from the literature based on the ability to optimise the objective functions. It was possible to do this for iMod, WeiMod, DynOptMod and SimMod as methods with the same objective functions exist. For iMod and WeiMod (Chapters 3 and 4 respectively) there were many comparison methods to chose from due to the multitude of existing modularity optimisation methods. The methods chosen represented a variety of optimisation techniques and were well-established methods. The performance of both iMod and WeiMod was demonstrated on synthetic as well as a number of real life networks. Similarly, a direct comparison for DynOptMod and SimMod in Chapter 7 was possible with the Average-method of Aynaud and Guillaume [25] on four dynamic networks. Overall, comparisons with associated methods from the literature showed that each method performed very well within specific limits. For example, iMod and WeiMod found the best solutions in the synthetic network analysis and for real networks with up to 1133 and 889 nodes respectively. DynOptMod guarantees globally optimal solutions and therefore for some small cases detected the best solutions but scalability soon became a problem. Similarly on certain networks SimMod found the best solutions, but for the dynamic network with the largest snapshot networks Average-method performed better.

  A direct comparison was more difficult for OverMod since results first depend on the algorithm used to find the hard partition, and second there are no other methods with an identical objective function. A method comparison with the

karate network was carried out in order to illustrate the large variability in results between overlapping clustering methods and indicated that a direct comparison between methods may not be very helpful but context specific evaluations are required instead, which is tackled in the final research aim below.

- *To demonstrate the potential of such methods to find meaningful results in biological applications.*

  The final research aim was addressed in Chapters 5 and 6. Functional enrichment analysis of modules detected by modularity optimisation is regularly carried out and has shown the appoach to find biologically meaningful results [43, 115, 119]. Therefore focus here was on analysing the less applied and relatively new approach of detecting overlapping modules. Several previous studies have assessed the functional enrichment of overlapping communities to illustrate that they are more biologically coherent than non-overlapping communities [185, 206, 223]. A different route was taken in this thesis; properties of the multi-clustered nodes were investigated in an attempt to characterise the type of node (protein, gene etc.) that is likely to anchor functional units. First, OverMod was shown to multi-cluster proteins in the human and rat PPI networks that were on average more highly connected and more multi-functional than mono-clustered proteins. The methodology was then applied to the integrated network of *Fusarium graminearum*. Again multi-clustered nodes were shown to be highly connected and involved in more biological functions but furthermore, were shown to contain more protein domains and to be enriched for nodes with 'connector' roles, according to definitions in [83]. The properties found reflected the idea that nodes which link modules do indeed play some strategic role in the overall functioning of the network. Therefore, these specific applications demonstrated that OverMod could find biologically meaningful results, therefore contributing to the achievement of this research aim. However future work will be largely aimed at more applications to thoroughly investigate the ability of OverMod to detect 'important' nodes.

8.3 Contributions

Key contributions of this thesis are listed below.

- Extended the use of an existing mathematical programming approach to find disjoint communities in medium to large sized networks.
- Extended the use of mathematical programming models to find disjoint partitions of weighted networks.
- Developed a two-stage mathematical programming procedure to transform disjoint to overlapping communities.
- Extended two mathematical programming models to simultaneously cluster multiple network snapshots.
- Demonstrated that the procedure for detecting overlapping communities could multi-cluster proteins with properties indicating their topological and functional importance through biological and agriculture case studies.

8.4 Limitations

It has been previously acknowledged that modularity optimisation suffers from three limitations: NP-hardness [35], a resolution limit [73] and degeneracy of solutions [76]. These limitations are briefly discussed below.

- NP-hardness affects the scalability of the modularity-based methods presented in this thesis and as is mentioned below in Section 8.5, this will be addressed in future work. The NP-hardness of modularity optimisation ultimately results in a trade-off being made between accuracy and efficiency of clustering methods and as such, methods should be chosen according to experiment-specific requirements.
- The resolution limit in modularity optimisation results in the failure to detect communities that are smaller than a certain limit [73]. Although it may be possible that a module of any size comprises smaller modules, it is found that it is more likely to be the case if the number of intra-community links is less than or equal to $\sqrt{2L}$ where L is the total number of links in the network. A few methods have been proposed to overcome this including: an alternative objective function known as 'modularity density' [121], recursive partitioning of sub-networks [176, 218], re-weighting of network interactions [32] and multi-resolution methods that incorporate a tuneable parameter in the modularity objective function

[22, 119, 170, 171, 197]. However, it has recently been shown that the resolution limit also exists for multi-resolution techniques and it is in fact likely to be a general problem for all methods based on global optimisation [108]. This could indicate that in general, alternative methods may also exhibit some form of disadvantage or limitation as of yet undiscovered. Therefore, the aspect that leans in favour of modularity optimisaiton is that it is well-established and whereas new methods have not as yet been evaluated as thoroughly in as many different contexts, modularity has already proven itself to be an informative measure.

- Modularity optimisation solutions have been found to suffer from degeneracy. It has been shown that partitions that have modularity values close to the optimal partition can show great structural variation [76]. Furthermore, as the number of modules in a partition increases, the number of possible close-to-optimal partitions grows at least exponentially. The degenerate solutions are a result of the modularity function not strongly penalising partitions that merge modules. As the number of modules increases, the penalty (change in Q) for merging two modules vanishes. Consequently, it is unclear which of these equally 'good' or probable partitions is the correct or more meaningful without any external information about the network other than purely based on the interactions. However, less emphasis is put on the degeneracy of solutions in the clustering literature compared with the interest in the resolution limit and therefore, for the same reasons as above, modularity optimisation should not be dismissed due to this limitation.

The above limitations have raised doubts about the ability of modularity-based approaches to cluster complex networks. However, modularity optimisation has proven its utility and is well-established and commonly used. In particular, as has been indicated at several points throughout this thesis, clustering methods based on modularity optimisation have found informative results in biological applications. Therefore, as long as one is aware of the existence of the limitations when interpreting results, it is still reasonable to continue to invest time in developing modularity-based clustering methods. Overall, regarding the methodologies proposed in this thesis, their true advantages and disadvantages will be uncovered in more context-specific applications.

8.5 Future work

Future work will involve the extension and improvement of the methodologies presented in this thesis. One of the main outcomes of the method evaluations was that scalability is a problem for most of the proposed models. Biological networks can range from a few hundred to a few thousand, or even tens of thousands of nodes. Since the clustering procedures in this thesis have been developed with bioinformatics applications in mind, increasing method efficiency is required. For WeiMod there are two avenues for possible future work: (i) generalise the MIQP formulation of modularity optimisation and make a fully generalised iMod procedure for clustering weighted networks or (ii) focus on reducing the computational cost of WeiMod. Despite iMod having been shown to be more accurate than WeiMod on unweighted networks, the former is not necessarily the preferable option as results in Chapter 3 identified that Stage 2 of the iMod method is often unstable in terms of computational cost. Therefore efforts may be better directed at improving the efficiency of WeiMod.

Contributions to the improvement of the scalability of WeiMod may come from using alternative MINLP solvers. Several are available through GAMS [6]. This is touched on briefly in Chapter 4 where a comparison is made between SBB and DICOPT. For these two solvers, results did not show much variation, however future work will further investigate alternative solvers, as well as explore the possibility of combining solvers. In addition, the effect of parallel processing can be investigated. Further improvements to scalability may come from including symmetry breaking constraints in order to avoid equivalent solutions, such as those seen in OptMod [219]. The scalability limitations of WeiMod and the corresponding future work apply equally to SimMod. However, the added consideration with dynamic networks is the effect of the number of snapshots as well as the number of nodes in each of the snapshots. Therefore resolving scalability issues for SimMod may prove a more difficult task.

Future work will also involve the continuation of the method development that has been presented in this thesis, i.e. continuing the seach for more realistic community structure modelling frameworks. For example in Chapter 4, Section 4.5 objective functions are suggested to (i) cluster directed networks and (ii) overcome the resolution limit. The proposed measures can easily replace standard modularity in the WeiMod mathematical model.

Furthermore, at the end of Chapter 7, the idea of evolutionary clustering is suggested as the next solution to clustering dynamic networks. Formally, an evolutionary clustering algorithm is given the current state and the previous state of a dynamic network, G_t and G_{t-1} respectively and a partition of the network at time $t-1$ as input and returns a partition of the current network at time t. This problem statement can be formulated as a mathematical model where the objective function comprises the modularity measure of the network at the current time point plus a preservation coefficient, measuring the proportion of node pairs occurring together in the same module at time t and in the partition of time $t-1$. Additionally, ways in which the evolution of a module can be tracked across network snapshots should also be investigated.

Moreover, this is not the end of the method development; future directions are still to be considered. A possible option is to incorporate prior biological knowedge into the models in order to achieve more accurate solutions. For example, nodes with similar functional annotations could be constrained to be in the same community [112]. Such constraints could be easily implemented within the mathematical programming framework that has been employed throughout the thesis.

Biological applications will also feature in future work in order to first, evaluate more robustly the cababilities of the methodologies, but also to use the methods to generate hypotheses which can possibly be validated in experimental investigations.

More specifically, future work involving OverMod will investigate features of multi-clustered nodes, other than those considered in Chapters 5 and 6, that reinforce their role as connectors and their influential position. For example, for the human PPI network, it can be investigated whether multi-clustered proteins are enriched for druggable targets according to the druggable genome [177] as has been done in [223]. This will again indicate whether OverMod has the potential to identify important nodes, such as drug targets.

Furthermore, the application pipeline outlined in Chapter 6 can be executed on better annotated organisms. If more virulence associated genes are known, or equally a significant number of genes with other important properties, the analysis may arrive at more concrete conclusions.

Finally, the biological significance of clustering dynamic networks will be explored. As this area of investigation is still relatively in its infancy, it is not yet clear what form such future work will take. However questions that will be considered include: (i) what

do the modules of a consensus partition represent, are they indeed enriched for the most important biological functions in the process being modelled and (ii) does community structure change over time, and if so what biological changes do the topological changes correspond to? The clustering of dynamic biological networks presents a large area of consideration for future work.

8.6 Concluding remarks

This thesis has advanced the use of mathematical programming in community structure detection. Several stages of an evolving problem statement have been addressed resulting in more accurate modelling frameworks for the detection of communities in complex networks. Furthermore, the proposed methodologies show improvements on similar existing approaches.

Additionally, these methodological developments go a long way to represent many of the intricate relationships that exist in complex biological systems. Evidence of this can been seen in applications in biological and agricultural case studies. The promising results indicate much potential for future appliations.

Chapter 9

Bibliography

[1] http://www.gams.com/dd/docs/solvers/sbb.pdf.

[2] http://vlado.fmf.uni-lj.si/pub/networks/data/GD/GD.htm.

[3] http://vlado.fmf.uni-lj.si/pub/networks/data/map/USAir97.net.

[4] http://www.gams.com/dd/docs/solvers/conopt.pdf.

[5] http://www.gams.com/dd/docs/solvers/dicopt.pdf.

[6] http://www.gamsworld.org/minlp/solvers.htm.

[7] http://www.ebi.ac.uk/GOA/rat_release.html.

[8] http://www.ebi.ac.uk/GOA/human_release.html.

[9] www.fao.org.

[10] http://www.broadinstitute.org/news/117.

[11] ftp://ftpmips.gsf.de/FGDB/v32.

[12] http://mips.helmholtz-muenchen.de/genre/proj/FGDB/.

[13] ILOG, ILOG CPLEX10.0 User's Manual, 2006.

[14] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *Eur. Phys. J. B*, 66:409–418, 2008.

[15] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.

[16] R. Albert. Scale-free networks in cell biology. *Journal of Cell Science*, 118:4947–4957, 2005.

[17] R. Albert, H. Jeong, and A. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

[18] M. Aldana and P. Cluzel. A natural class of robust networks. *Proceedings of the National Academy of Sciences*, 100:8710–8714, 2003.

[19] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, S. Perron, and L. Liberti. Column generation algorithms for exact modularity maximization in networks. *Physical Review E*, 82:046112, 2010.

[20] U. Alon. `http://www.weizmann.ac.il/mcb/UriAlon/`.

[21] A. Arenas, A. Daz-Guilera, and C. J. Prez-Vicente. Synchronization processes in complex networks. *Physica D: Nonlinear Phenomena*, 224:27–34, 2006.

[22] A. Arenas, A. Fernandez, and S. Gomez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10:053039, 2008.

[23] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25:25–29, 2000.

[24] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Trans. Knowl. Discov. Data*, 3:16:1–16:36, 2009.

[25] T. Aynaud and J.-L. Guillaume. Multi-Step Community Detection and Hierarchical Time Segmentation in Evolving Networks. In *Fifth SNA-KDD Workshop Social Network Mining and Analysis, in conjunction with the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*, 2011.

[26] N. Azizifard, M. Mahdavi, and B. Nasersharif. Modularity optimization for clustering insocial networks. *International Conference on Emerging Trends in Computer and Image Processing (ICETCIP2011)*, Bangkok, Thailand, December 2011.

[27] G. D. Bader, I. Donaldson, C. Wolting, B. F. F. Ouellette, T. Pawson, and C. W. V. Hogue. Bind-the biomolecular interaction network database. *Nucleic Acids Research*, 29:242–245, 2001.

[28] A.-L. Barabsi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[29] T. Barrett and R. Edgar. Mining microarray data at NCBI's Gene Expression Omnibus (GEO)*. *Methods in molecular biology*, 338:175–190, 2006.

[30] E. Becker, B. Robisson, C. E. Chapple, A. Guénoche, and C. Brun. Multifunctional proteins revealed by overlapping clustering in protein interaction network. *Bioinformatics*, 28:84–90, 2012.

[31] L. Bennett, A. Lysenko, L. G. Papageorgiou, M. Urban, K. Hammond-Kosack, C. Rawlings, M. Saqi, and S. Tsoka. Detection of multi-clustered genes and community structure for the plant pathogenic fungus fusarium graminearum. CSMB 2012, pages 69–86, London, UK, UK, 2012. LNCS.

[32] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips. Tolerating the community detection resolution limit with edge weighting. *Physical Review E*, 83:056119, 2011.

[33] R. Blasco, S. Francoz, D. Santamaría, M. Cañamero, P. Dubus, J. Charron, M. Baccarini, and M. Barbacid. c-raf, but not b-raf, is essential for development of k-ras oncogene-driven non-small cell lung carcinoma. *Cancer Cell*, 19:652–63, 2011.

[34] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008, 2008.

[35] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20:172 –188, 2008.

[36] S. Cafieri, P. Hansen, and L. Liberti. Locally optimal heuristic for modularity maximization of networks. *Physical Review E*, 83:056105, 2011.

[37] Y. Cai, C. Shi, Y. Dong, Q. Ke, and B. Wu. A novel genetic algorithm for overlapping community detection. In *Proceedings of the 7th international conference on Advanced Data Mining and Applications - Volume Part I*, ADMA'11, pages 97–108, Berlin, Heidelberg, 2011. Springer-Verlag.

[38] M. A. Caligiuri, R. Briesewitz, J. Yu, L. Wang, M. Wei, K. J. Arnoczky, T. B. Marburger, J. Wen, D. Perrotti, C. D. Bloomfield, and S. P. Whitman. Novel c-cbl and cbl-b ubiquitin ligase mutations in human acute myeloid leukemia. *Blood*, 110:1022–1024, 2007.

[39] R. Caspi, T. Altman, J. M. Dale, K. Dreher, C. A. Fulcher, F. Gilham, P. Kaipa, A. S. Karthikeyan, A. Kothari, M. Krummenacker, M. Latendresse, L. A. Mueller, S. Paley, L. Popescu, A. Pujar, A. G. Shearer, P. Zhang, and P. D. Karp. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research*, 38:D473–D479, 2010.

[40] T. F. C. . R. O. S. Center. The transcriptional network that controls growth arrest and differentiation in a human myeloid leukemia cell line. *Nature Genetics*, 41:553 – 562, 2009.

[41] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 554–560, New York, NY, USA, 2006. ACM.

[42] D. Chen, M. Shang, Z. Lv, and Y. Fu. Detecting overlapping communities of weighted networks via a local algorithm. *Physica A: Statistical Mechanics and its Applications*, 389:4177 – 4187, 2010.

[43] J. Chen and B. Yuan. Detecting functional modules in the yeast proteinprotein interaction network. *Bioinformatics*, 22:2283–2290, 2006.

[44] H. Cheng, Y. Zhou, X. Huang, and J. Yu. Clustering large attributed information networks: an efficient incremental computing approach. *Data Mining and Knowledge Discovery*, 25:450–477, 2012.

[45] J. M. Cherry, E. L. Hong, C. Amundsen, R. Balakrishnan, G. Binkley, E. T. Chan, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. R. Engel, D. G. Fisk,

J. E. Hirschman, B. C. Hitz, K. Karra, C. J. Krieger, S. R. Miyasato, R. S. Nash, J. Park, M. S. Skrzypek, M. Simison, S. Weng, and E. D. Wong. Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic Acids Research*, 40:D700–D705, 2012.

[46] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 153–162, New York, NY, USA, 2007. ACM.

[47] N. Chia and N. Goldenfeld. Statistical Mechanics of Horizontal Gene Transfer in Evolutionary Ecology. *Journal of Statistical Physics*, 142:1287–1301, 2011.

[48] A. Clauset. `http://www.cs.unm.edu/~aaron/research/fastmodularity.htm`.

[49] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[50] V. Colizza, R. Pastor-Satorras, and A. Vespignani. Reaction-diffusion processes and metapopulation models in heterogeneous networks. *Nature Physics*, 3:276–282, 2007.

[51] C. A. Cuomo, U. Gldener, J.-R. Xu, F. Trail, B. G. Turgeon, A. Di Pietro, J. D. Walton, L.-J. Ma, S. E. Baker, M. Rep, G. Adam, J. Antoniw, T. Baldwin, S. Calvo, Y.-L. Chang, D. DeCaprio, L. R. Gale, S. Gnerre, R. S. Goswami, K. Hammond-Kosack, L. J. Harris, K. Hilburn, J. C. Kennell, S. Kroken, J. K. Magnuson, G. Mannhaupt, E. Mauceli, H.-W. Mewes, R. Mitterbauer, G. Muehlbauer, M. Mnsterktter, D. Nelson, K. O'Donnell, T. Ouellet, W. Qi, H. Quesneville, M. I. G. Roncero, K.-Y. Seong, I. V. Tetko, M. Urban, C. Waalwijk, T. J. Ward, J. Yao, B. W. Birren, and H. C. Kistler. The fusarium graminearum genome reveals a link between localized polymorphism and pathogen specialization. *Science*, 317:1400–1402, 2007.

[52] M. D'Amelio, V. Cavallucci, S. Middei, C. Marchetti, S. Pacioni, A. Ferri, A. Diamantini, D. De Zio, P. Carrara, L. Battistini, S. Moreno, A. Bacci, M. Ammassari-Teule, H. Marie, and F. Cecconi. Caspase-3 triggers early synaptic dysfunction in a mouse model of alzheimer's disease. *Nature Neuroscience*, 14:69–76, 2011.

[53] L. Dartnell, E. Simeonidis, M. Hubank, S. Tsoka, I. D. L. Bogle, and L. G. Papageorgiou. Robustness of the p53 network and biological hackers. *FEBS Letters*, 579:3037 – 3042, 2005.

[54] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti. Generalized louvain method for community detection in large networks. In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pages 88 –93, 2011.

[55] R. Dean, J. A. L. Van Kan, Z. A. Pretorious, K. E. Hammond-Kosack, A. Di Pietro, P. D. Spanu, J. J. Rudd, M. Dickman, R. Kahmann, J. Ellis, and G. D. Foster. The top 10 fungal pathogens in molecular plant pathology. *Molecular Plant Pathology*, 13:414–430, 2012.

[56] J. Diesner, T. L. Frantz, and K. M. Carley. Communication networks from the enron email corpus "it's always about the people. enron is no different". *Comput. Math. Organ. Theory*, 11:201–228, 2005.

[57] L. Donetti and M. A. Muñoz. Improved spectral algorithm for the detection of network communities. In *Proceedings of the 8th Granada Seminar - Computational and Statistical Physics*, pages 1–2, 2005.

[58] W. Du and S. Tan. Optimizing modularity to identify semantic orientation of chinese words. *Expert Systems with Applications*, 37:5094 – 5100, 2010.

[59] D. Duan, Y. Li, Y. Jin, and Z. Lu. Community mining on dynamic weighted directed graphs. In *Proceedings of the 1st ACM international workshop on Complex networks meet information; knowledge management*, CNIKM '09, pages 11–18, New York, NY, USA, 2009. ACM.

[60] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72:027104, 2005.

[61] C. Elkan. Using the triangle inequality to accelerate k-means. In *ICML'03*, pages 147–153, 2003.

[62] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30:1575–1584, 2002.

[63] S. Erten, X. Li, G. Bebek, J. Li, and M. Koyuturk. Phylogenetic analysis of modularity in protein interaction networks. *BMC Bioinformatics*, 10:333, 2009.

[64] E. Estrada and N. Hatano. Communicability graph and community structures in complex networks. *Applied Mathematics and Computation*, 214:500–511, 2009.

[65] Y. Fan, M. Li, P. Zhang, J. Wu, and Z. Di. Accuracy and precision of methods for community identification in weighted networks. *Physica A: Statistical Mechanics and its Applications*, 377:363 – 372, 2007.

[66] D. J. Fenn, M. A. Porter, M. McDonald, S. Williams, N. F. Johnson, and N. S. Jones. Dynamic communities in multichannel data: An application to the foreign exchange market during the 2007–2008 credit crisis. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19:033119, 2009.

[67] A. Fernndez. Molecular basis for evolving modularity in the yeast protein interaction network. *PLoS Comput Biol*, 3:e226, 2007.

[68] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identification of web communities. *Computer*, 35:66–71, 2002.

[69] S. Fortunato. `https://sites.google.com/site/santofortunato/mutual3.tar.gz?attredirects=0`.

[70] S. Fortunato. `https://sites.google.com/site/santofortunato/binary_networks.tar.gz?attredirects=0`.

[71] S. Fortunato. `https://sites.google.com/site/santofortunato/weighted_networks.tar.gz?attredirects=0`.

[72] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75 – 174, 2010.

[73] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104:36–41, 2007.

[74] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99:7821–7826, 2002.

[75] K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabsi. The human disease network. *Proceedings of the National Academy of Sciences*, 104:8685–8690, 2007.

[76] B. Good, Y. D. Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81:046106, 2010.

[77] R. Görke, P. Maillard, A. Schumm, C. Staudt, and D. Wagner. Dynamic Graph Clustering Combining Modularity and Smoothness. Technical report, ITI Wagner,

Department of Informatics, Karlsruhe Institute of Technology (KIT), 2011. Karlsruhe Reports in Informatics 2011-11 (and invitational submission to a Special Issue of the ACM Journal on Experimental Algorithmics).

[78] R. S. Goswami and H. C. Kistler. Heading for disaster: Fusarium graminearum on cereal crops. *Molecular Plant Pathology*, 5:515–525, 2004.

[79] S. Gregory. An algorithm to find overlapping community structure in networks. In *Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, pages 91–102, Berlin, Heidelberg, 2007. Springer-Verlag.

[80] S. Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12:103018, 2010.

[81] J. Guillaume. `https://sites.google.com/site/findcommunities/`.

[82] J.-L. Guillaume. `http://jlguillaume.free.fr/www/programs.php`.

[83] R. Guimera and L. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900, 2005.

[84] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Physical Review E*, 68:065103, 2003.

[85] R. Guimera, M. Sales-Pardo, and L. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70:025101, 2004.

[86] U. Gldener, M. Mnsterktter, G. Kastenmller, N. Strack, J. van Helden, C. Lemer, J. Richelles, S. J. Wodak, J. Garca-Martnez, J. E. Prez-Ortn, H. Michael, A. Kaps, E. Talla, B. Dujon, B. Andr, J. L. Souciet, J. De Montigny, E. Bon, C. Gaillardin, and H. W. Mewes. Cygd: the comprehensive yeast genome database. *Nucleic Acids Research*, 33:D364–D368, 2005.

[87] J. Hallinan. Gene duplication and hierarchical modularity in intracellular interaction networks. *Biosystems*, 74:51–62, 2004.

[88] M. B. Hastings. Community detection as an inference problem. *Physical Review E*, 74:035102, 2006.

[89] C. A. Heinlein and C. Chang. Androgen receptor in prostate cancer. *Endocrine Reviews*, 25:276–308, 2004.

[90] A. Hintze and C. Adami. Evolution of complex modular biological networks. *PLoS Comput Biol*, 4:e23, 2008.

[91] P. Holme, M. Huss, and H. Jeong. Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19:532–538, 2003.

[92] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5249–5253, 2004.

[93] H. Jeong, S. P. Mason, A.-L. Barabasi, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

[94] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407:651–654, 2000.

[95] D. Jin, B. Yang, C. Baquero, D. Liu, D. He, and J. Liu. A markov random walk under constraint for discovering overlapping communities in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011:P05031, 2011.

[96] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, 40:D109–D114, 2012.

[97] P. D. Karp, M. Riley, M. Saier, I. T. Paulsen, J. Collado-Vides, S. M. Paley, A. Pellegrini-Toole, C. Bonavides, and S. Gama-Castro. The ecocyc database. *Nucleic Acids Research*, 30:56–58, 2002.

[98] B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83:016107, 2011.

[99] M.-S. Kim and J. Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc. VLDB Endow.*, 2:622–633, 2009.

[100] E. S. Knudsen and J. Y. J. Wang. Targeting the rb-pathway in cancer therapy. *Clinical Cancer Research*, 16:1094–1099, 2010.

[101] D. E. Knuth. *The Stanford GraphBase: a platform for combinatorial computing.* ACM, New York, NY, USA, 1993.

[102] V. Krebs. `http://www.orgnet.com/`.

[103] S. Kühner, V. van Noort, M. J. Betts, A. Leo-Macias, C. Batisse, M. Rode, T. Yamada, T. Maier, S. Bader, P. Beltran-Alvarez, D. Castaño Diez, W.-H. Chen, D. Devos, M. Güell, T. Norambuena, I. Racke, V. Rybin, A. Schmidt, E. Yus, R. Aebersold, R. Herrmann, B. Böttcher, A. S. Frangakis, R. B. Russell, L. Serrano, P. Bork, and A.-C. Gavin. Proteome Organization in a Genome-Reduced Bacterium. *Science*, 326:1235–1240, 2009.

[104] J. Khler, J. Baumbach, J. Taubert, M. Specht, A. Skusa, A. Regg, C. Rawlings, P. Verrier, and S. Philippi. Graph-based analysis and visualization of experimental results with ondex. *Bioinformatics*, 22:1383–1390, 2006.

[105] S. Lab. `http://etseq.urv.cat/seeslab/downloads/`.

[106] D. Lai, X. Wu, H. Lu, and C. Nardini. Learning overlapping communities in complex networks via non-negative matrix factorization. *International Journal of Modern Physics C*, 22:1173–1190, 2011.

[107] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80:016118, 2009.

[108] A. Lancichinetti and S. Fortunato. Limits of modularity maximization in community detection. *Physical Review E*, 84:066122, 2011.

[109] A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Scientific Reports*, 2:336, 2012.

[110] A. Lancichinetti, S. Fortunato, and J. Kertsz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11:033015, 2009.

[111] T. Lastusilta, M. R. Bussieck, and T. Westerlund. Comparison of some high-performance minlp solvers. *Chem. Eng. Trans*, 11:125–130, 2007.

[112] A. J. Lee, M.-C. Lin, and C.-M. Hsu. Mining dense overlapping subgraphs in weighted protein-protein interaction networks. *Biosystems*, 103:392 – 399, 2011.

[113] I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306:1555–1558, 2004.

[114] I. Lee and E. M. Marcotte. Integrating functional genomics data. In J. M. Keith and J. M. Walker, editors, *Bioinformatics*, volume 453 of *Methods in Molecular Biology*, pages 267–278. Humana Press, 2008.

[115] J. Lee, S. P. Gross, and J. Lee. Extraction of hidden information by efficient community detection in networks. 2012. arXiv/1209.2873.

[116] J. Lee, S. P. Gross, and J. Lee. Modularity optimization by conformational space annealing. *Physical Review E*, 85:056702, 2012.

[117] E. A. Leicht and M. E. J. Newman. Community Structure in Directed Networks. *Physical Review Letters*, 100:118703, 2008.

[118] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6:29–123, 2009.

[119] A. Lewis, N. Jones, M. Porter, and C. Deane. The function of communities in protein interaction networks at multiple scales. *BMC Systems Biology*, 4:100, 2010.

[120] H.-J. Li, Y. Wang, L.-Y. Wu, J. Zhang, and X.-S. Zhang. Potts model based on a markov process computation solves the community structure problem effectively. *Physical Review E*, 86:016109, 2012.

[121] Z. Li, S. Zhang, R. S. Wang, X. S. Zhang, and L. Chen. Quantitative function for community detection. *Physical Review E*, 77:036109, 2008.

[122] M. Lima. `http://www.mslima.com/mfadt/thesis/2004/08/protect-hubs_18.html`.

[123] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Analyzing communities and their evolutions in dynamic social networks. *ACM Trans. Knowl. Discov. Data*, 3:1–31, 2009.

[124] R. Litman, R. Gupta, R. M. J. Brosh, and S. B. Cantor. Brca-fa pathway as a target for anti-tumor drugs. *Anticancer Agents Med Chem*, 8:426430, 2008.

[125] G. Liu, L. Wong, and H. N. Chua. Complex discovery from weighted ppi networks. *Bioinformatics*, 25:1891–1897, 2009.

[126] X. Liu, W.-H. Tang, X.-M. Zhao, and L. Chen. A network approach to predict pathogenic genes for *fusarium graminearum*. *PLoS ONE*, 5:e13021, 2010.

[127] J. Long and D. Lahiri. Current drug targets for modulating alzheimer's amyloid precursor protein: role of specific micro-rna species. *Curr Med Chem*, 18:3314–3321, 2011.

[128] D. M. Lorenz, A. Jeng, and M. W. Deem. The emergence of modularity in biological systems. *Physics of Life Reviews*, 8:129 – 160, 2011.

[129] S. Lu, J. Lee, M. Revelo, X. Wang, S. Lu, and Z. Dong. Smad3 is overexpressed in advanced human prostate cancer and necessary for progressive growth of prostate cancer cells in nude mice. *Clinical Cancer Research*, 13:5692–5702, 2007.

[130] D. Lusseau. The emergent properties of a dolphin social network. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270:S186–S188, 2003.

[131] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: Can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54:pp. 396–405, 2003.

[132] A. Lysenko, M. Defoin-Platel, K. Hassani-Pak, J. Taubert, C. Hodgman, C. Rawlings, and M. Saqi. Assessing the functional coherence of modules found in multiple-evidence networks from arabidopsis. *BMC Bioinformatics*, 12:203, 2011.

[133] A. Lysenko, M. Urban, L. Bennett, S. Tsoka, E. Janowska-Sejda, C. Rawlings, K. Hammond-Kosack, and M. Saqi. Network-based data integration for predicting virulence proteins in the cereal infecting fungus *fusarium graminearum*. (submitted), 2012.

[134] X. Ma, L. Gao, and X. Yong. Eigenspaces of networks reveal the overlapping and hierarchical community structure more precisely. *Journal of Statistical Mechanics: Theory and Experiment*, 2010:P08012, 2010.

[135] E. Marcora and M. B. Kennedy. The huntington's disease mutation impairs huntingtin's role in the transport of nf-b from the synapse to the nucleus. *Human Molecular Genetics*, 19:4373–4384, 2010.

[136] E. Massaro, A. Guazzini, F. Bagnoli, and P. Liò. Information dynamics algorithm for detecting communities in networks. 2011. CoRR abs/1112.1224.

[137] A. Medus, G. Acuna, and C. Dorso. Detection of community structures in networks via global optimization. *Physica A: Statistical Mechanics and its Applications*, 358:593–604, 2005.

[138] S. Milgram. The small world problem. *Psychology Today*, 61:60–67, 1967.

[139] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[140] S. Ming-Sheng, C. Duan-Bing, and Z. Tao. Detecting overlapping communities based on community cores in complex networks. *Chinese Physics Letters*, 27:058901, 2010.

[141] J. Nacher, M. Hayashida, and T. Akutsu. Emergence of scale-free distribution in proteinprotein interaction networks based on random selection of interacting domain pairs. *Biosystems*, 95:155 – 159, 2009.

[142] R. R. Nadakuditi and M. E. J. Newman. Graph spectra and the detectability of community structure in networks. *Phys. Rev. Lett.*, 108:188701, 2012.

[143] T. Narayanan, M. Gersten, S. Subramaniam, and A. Grama. Modularity detection in protein-protein interaction networks. *BMC Research Notes*, 4:569, 2011.

[144] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98:404–409, 2001.

[145] M. E. J. Newman. Analysis of weighted networks. *Phys Rev E*, 70:056131, 2004.

[146] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[147] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103:8577–8582, 2006.

[148] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[149] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104:9564–9569, 2007.

[150] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849–856. MIT Press, 2001.

[151] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai. Overlapping communities in dynamic networks: their detection and mobile applications. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, MobiCom '11, pages 85–96, New York, NY, USA, 2011. ACM.

[152] N. P. Nguyen, T. N. Dinh, Y. Xuan, and M. T. Thai. Adaptive algorithms for detecting community structure in dynamic social networks. In *INFOCOM'11*, pages 2282–2290, 2011.

[153] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending modularity definition for directed graphs with overlapping communities. *Journal Of Statistical Mechanics: Theory And Experiment*, 3:03024, 2008.

[154] T. Obayashi, K. Kinoshita, K. Nakai, M. Shibaoka, S. Hayashi, M. Saeki, D. Shibata, K. Saito, and H. Ohta. Atted-ii: a database of co-expressed genes and cis elements for identifying co-regulated gene groups in arabidopsis. *Nucleic Acids Research*, 35:D863–D869, 2007.

[155] M. Olivier, M. Hollstein, and P. Hainaut. Tp53 mutations in human cancers: Origins, consequences, and clinical use. *Cold Spring Harbor Perspectives in Biology*, 2:a001008, 2010.

[156] T. Opsahl. *Structure and Evolution of Weighted Networks*. University of London (Queen Mary College), London, UK, 2009.

[157] M. Oti and H. Brunner. The modular nature of genetic diseases. *Clinical Genetics*, 71:1–11, 2007.

[158] G. Palla, A.-L. Barabasi, and T. Vicsek. Quantifying social group evolution. *Nature*, 446:664–667, 2007.

[159] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[160] H. Parkinson, M. Kapushesky, N. Kolesnikov, G. Rustici, M. Shojatalab, N. Abeygunawardena, H. Berube, M. Dylag, I. Emam, A. Farne, E. Holloway, M. Lukk,

J. Malone, R. Mani, E. Pilicheva, T. F. Rayner, F. Rezwan, A. Sharma, E. Williams, X. Z. Bradley, T. Adamusiak, M. Brandizi, T. Burdett, R. Coulson, M. Krestyaninova, P. Kurnosov, E. Maguire, S. G. Neogi, P. Rocca-Serra, S.-A. Sansone, N. Sklyar, M. Zhao, U. Sarkans, and A. Brazma. Arrayexpress updatefrom an archive of functional genomics experiments to the atlas of gene expression. *Nucleic Acids Research*, 37:D868–D872, 2009.

[161] C. Pizzuti. Overlapped community detection in complex networks. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pages 859–866, New York, NY, USA, 2009. ACM.

[162] P. Pons and M. Latapy. Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, 10:191–218, 2006.

[163] T. M. Przytycka, M. Singh, and D. K. Slonim. Toward the dynamic interactome: it's about time. *Briefings in Bioinformatics*, 11:15–29, 2010.

[164] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[165] P. B. Rainey and T. F. Cooper. Evolution of bacterial diversity and the origins of modularity. *Research in Microbiology*, 155:370 – 375, 2004.

[166] M. J. Rattigan, M. Maier, and D. Jensen. Using structure indices for efficient approximation of network properties. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 357–366, New York, NY, USA, 2006. ACM.

[167] P. K. Reddy, M. Kitsuregawa, P. Sreekanth, and S. S. Rao. A graph based approach to extract a neighborhood customer community for collaborative filtering. In *Proceedings of the Second International Workshop on Databases in Networked Information Systems*, DNIS '02, pages 188–200, London, UK, UK, 2002. Springer-Verlag.

[168] J. Reichardt and S. Bornholdt. Detecting fuzzy community structures in complex networks with a potts model. *Phys. Rev. Lett.*, 93:218701, 2004.

[169] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74:016110, 2006.

[170] J. Reichardt and S. Bornholdt. When are networks truly modular? *Physica D: Nonlinear Phenomena*, 224:20 – 26, 2006.

[171] P. Ronhovde and Z. Nussinov. Local resolution-limit-free Potts model for community detection. *Physical Review E*, 81:046114, 2010.

[172] R. Rosenthal. *GAMS - A user's guide*. GAMS Development Corporation, Washington D.C., USA, 2008.

[173] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104:7327–7331, 2007.

[174] M. Rosvall and C. T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PLoS ONE*, 6:e18209, 2011.

[175] J. Ruan. `http://www.cs.utsa.edu/~jruan/Software.html`.

[176] J. Ruan and W. Zhang. Identifying network communities with a high resolution. *Physical Review E*, 77:1–12, 2008.

[177] A. P. Russ and S. Lampel. The druggable genome: an update. *Drug Discovery Today*, 10:1607 – 1610, 2005.

[178] C. J. Ryan, A. Roguev, K. Patrick, J. Xu, H. Jahari, Z. Tong, P. Beltrao, M. Shales, H. Qu, S. R. Collins, J. I. Kliegman, L. Jiang, D. Kuo, E. Tosti, H.-S. Kim, W. Edelmann, M.-C. Keogh, D. Greene, C. Tang, P. Cunningham, K. M. Shokat, G. Cagney, J. P. Svensson, C. Guthrie, P. J. Espenshade, T. Ideker, and N. J. Krogan. Hierarchical modularity and the evolution of genetic interactomes across species. *Molecular Cell*, 46:691 – 704, 2012.

[179] M. Salath and J. H. Jones. Dynamics and control of diseases in networks with community structure. *PLoS Comput Biol*, 6:e1000736, 2010.

[180] P. Schuetz and A. Caflisch. Multistep greedy algorithm identifies community structure in real-world and computer-generated networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 78:026112, 2008.

[181] M. Schuldiner, S. R. Collins, N. J. Thompson, V. Denic, A. Bhamidipati, T. Punna, J. Ihmels, B. Andrews, C. Boone, J. F. Greenblatt, J. S. Weissman, N. J, and

D. O. M. Genetics. Exploration of the function and organization of the yeast early secretory pathway through an epistatic miniarray profile. *Cell*, 123:507–519, 2005.

[182] H.-W. Shen, X.-Q. Cheng, and J.-F. Guo. Quantifying and identifying the overlapping community structure in networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009:P07042, 2009.

[183] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31:64–8, 2002.

[184] J. Shetty and J. Adibi. `http://www.isi.edu/~adibi/Enron/Enron.htm`.

[185] Y.-K. Shih and S. Parthasarathy. Identifying functional modules in interaction networks through overlapping markov clustering. *Bioinformatics*, 28:i473–i479, 2012.

[186] N. Simonis, J.-F. Rual, A.-R. Carvunis, M. Tasan, I. Lemmens, T. Hirozane-Kishikawa, T. Hao, J. M. Sahalie, K. Venkatesan, F. Gebreab, S. Cevik, N. Klitgord, C. Fan, P. Braun, N. Li, N. Ayivi-Guedehoussou, E. Dann, N. Bertin, D. Szeto, A. Dricot, M. A. Yildirim, C. Lin, A.-S. de Smet, H.-L. Kao, C. Simon, A. Smolyar, J. S. Ahn, M. Tewari, M. Boxem, S. Milstein, H. Yu, M. Dreze, J. Vandenhaute, K. C. Gunsalus, M. E. Cusick, D. E. Hill, J. Tavernier, F. P. Roth, and M. Vidal. Empirically controlled mapping of the *Caenorhabditis elegans* protein-protein interactome network. *Nat Methods*, 6:47–54, 2008.

[187] M. E. Smoot, K. Ono, J. Ruscheinski, P.-L. Wang, and T. Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27:431–432, 2011.

[188] H. Son, Y.-S. Seo, K. Min, A. R. Park, J. Lee, J.-M. Jin, Y. Lin, P. Cao, S.-Y. Hong, E.-K. Kim, S.-H. Lee, A. Cho, S. Lee, M.-G. Kim, Y. Kim, J.-E. Kim, J.-C. Kim, G. J. Choi, S.-H. Yun, J. Y. Lim, M. Kim, Y.-H. Lee, Y.-D. Choi, and Y.-W. Lee. A phenome-based functional analysis of transcription factors in the cereal head blight fungus, *fusarium graminearum*. *PLoS Pathog*, 7:e1002310, 2011.

[189] V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, 100:12123–12128, 2003.

[190] O. D. Staples, R. J. C. Steele, and S. Lain. p53 as a therapeutic target. *Surgeon*, 6:240243, 2008.

[191] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34:D535–D539, 2006.

[192] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 687–696, New York, NY, USA, 2007. ACM.

[193] Z. Sun, J. Zheng, and H. Hu. Finding community structure in spatial maritime shiping networks. *International Journal of Modern Physics C*, 23:1250044, 2012.

[194] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, L. J. Jensen, and C. v. Mering. The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39:D561–D568, 2011.

[195] R. Tanaka. Scale-rich metabolic networks. *Phys. Rev. Lett.*, 94:168101, 2005.

[196] X. Tang, J. Wang, B. Liu, M. Li, G. Chen, and Y. Pan. A comparison of the functional modules identified from time course and static ppi network data. *BMC Bioinformatics*, 12:339, 2011.

[197] V. A. Traag, P. Van Dooren, and Y. Nesterov. Narrow scope for resolution-limit-free community detection. *Physical Review E*, 84:016114, 2011.

[198] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman. Communities and technologies. chapter Email as spectroscopy: automated discovery of community structure within organizations, pages 81–96. Kluwer, B.V., Deventer, The Netherlands, The Netherlands, 2003.

[199] UniProt Consortium. Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic acids research*, 40:D71–D75, 2012.

[200] S. M. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, The Netherlands, 2000.

[201] K. Voevodski, S.-H. Teng, and Y. Xia. Finding local communities in protein networks. *BMC Bioinformatics*, 10:297, 2009.

[202] A. Wagner and D. A. Fell. The small world inside large metabolic networks. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268:1803–1810, 2001.

[203] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks: [extended abstract]. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 1275–1276, New York, NY, USA, 2007. ACM.

[204] J. Wang, M. Li, Y. Deng, and Y. Pan. Recent advances in clustering methods for protein interaction networks. *BMC genomics*, 11:S10, 2010.

[205] X. Wang, L. Jiao, and J. Wu. Adjusting from disjoint to overlapping community detection of complex networks. *Physica A: Statistical Mechanics and its Applications*, 388:5045 – 5056, 2009.

[206] X. Wang, L. Li, and Y. Cheng. An overlapping module identification method in protein-protein interaction networks. *BMC Bioinformatics*, 13:S4, 2012.

[207] Y.-Y. Wang, J. C. Nacher, and X.-M. Zhao. Predicting drug targets based on protein domains. *Mol. BioSyst.*, 8:1528–1534, 2012.

[208] Z. Wang and J. Zhang. In search of the biological significance of modular structures in protein networks. *PLoS Computational Biology*, 3:e107, 2007.

[209] L. N. Z. L. X. W. F. H. Wang. Q, Su. L. Cyclin dependent kinase 1 inhibitors: a review of recent progress. *Current Medicinal Chemistry*, 18:2025–2043, 2011.

[210] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[211] F. Wei, W. Qian, C. Wang, and A. Zhou. Detecting overlapping community structures in networks. *World Wide Web*, 12:235–261, 2009.

[212] R. Winnenburg, T. K. Baldwin, M. Urban, C. Rawlings, J. Khler, and K. E. Hammond-Kosack. Phi-base: a new database for pathogen host interactions. *Nucleic Acids Research*, 34:459–464, 2006.

[213] R. Winnenburg, M. Urban, A. Beacham, T. K. Baldwin, S. Holland, M. Lindeberg, H. Hansen, C. Rawlings, K. E. Hammond-Kosack, and J. Khler. Phi-base update: additions to the pathogenhost interaction database. *Nucleic Acids Research*, 36:D572–D576, 2008.

[214] H. L. S. L. C. E. D. J. Wise RP, Caldo RA. Barleybase/plexdb. *Methods Mol Biol.*, 406:347–363, 2007.

[215] K. Wu, Y. Taki, K. Sato, Y. Sassa, K. Inoue, R. Goto, K. Okada, R. Kawashima, Y. He, A. C. Evans, and H. Fukuda. The overlapping community structure of structural brain network in young healthy individuals. *PLoS ONE*, 6:e19608, 2011.

[216] Z. Wu, Y. Lin, H. Wan, S. Tian, and K. Hu. Efficient overlapping community detection in huge real-world networks. *Physica A: Statistical Mechanics and its Applications*, 391:2475 – 2490, 2012.

[217] I. Xenarios, L. Salwnski, X. J. Duan, P. Higney, S.-M. Kim, and D. Eisenberg. Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30:303–305, 2002.

[218] G. Xu, L. Bennett, L. Papageorgiou, and S. Tsoka. Module detection in complex networks using integer optimisation. *Algorithms for Molecular Biology*, 5:36, 2010.

[219] G. Xu, S. Tsoka, and L. G. Papageorgiou. Finding community structures in complex networks using mixed integer optimisation. *The European Physical Journal B*, 60:231–239, 2007.

[220] Z. Xu, V. Tresp, A. Rettinger, and K. Kersting. K.: Social network mining with nonparametric relational models. In *Advances in Social Network Mining and Analysis - the Second SNA-KDD Workshop at KDD*, 2008.

[221] J. Yang, R. Wahdan-Alaswad, and D. Danielpour. Critical role of smad2 in tumor suppression and transforming growth factor-induced apoptosis of prostate epithelial cells. *Cancer Research*, 69:2185–2190, 2009.

[222] L. Ye, Y. He, H. Ye, X. Liu, L. Yang, Z. Cao, and K. Tang. Pathway-pathway network-based study of the therapeutic mechanisms by which salvianolic acid b regulates cardiovascular diseases. *Chinese Science Bulletin*, 57:1672–1679, 2012.

[223] Q. Yu, G.-H. H. Li, and J.-F. F. Huang. MOfinder: a novel algorithm for detecting overlapping modules from protein-protein interaction network. *Journal of biomedicine & biotechnology*, 2012:103702, 2012.

[224] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[225] M. Zarei, D. Izadi, and K. A. Samani. Detecting overlapping community structure of networks based on vertexvertex correlations. *Journal of Statistical Mechanics: Theory and Experiment*, 2009:P11013, 2009.

[226] S. Zhang, R.-S. Wang, and X.-S. Zhang. Uncovering fuzzy community structure in complex networks. *Physical Review E*, 76:046103, 2007.

[227] Z. Zhang and J. Zhang. A big world inside small-world networks. *PLoS ONE*, 4:e5686, 2009.

[228] L. Zhao, T. Liu, and J. Liu. Community detection in sample networks generated from gaussian mixture model. In Y. Tan, Y. Shi, Y. Chai, and G. Wang, editors, *Advances in Swarm Intelligence*, volume 6729 of *Lecture Notes in Computer Science*, pages 183–190. Springer Berlin / Heidelberg, 2011.

[229] X.-M. Zhao, X.-W. Zhang, W.-H. Tang, and L. Chen. Fppi: Fusarium graminearum protein-protein interaction database. *Journal of Proteome Research*, 8:4714–4721, 2009.

[230] G. Zinman, S. Zhong, and Z. Bar-Joseph. Biological interaction networks are conserved at the module level. *BMC Systems Biology*, 5:134, 2011.

List of Abbreviations

AIC-MICA	Average Information Content of the Most Informative Common Ancestor
ALL GO	GO terms from all three domains; MF, BP and CC
BB	Branch and Bound
BC	Belonging Coefficient
BIND	The Biomolecular Interaction Network Database
BioGRID	The Biological General Repository for Interaction Datasets
BMI	Between-Module-interactions
BP	Biological Process (GO domain)
CC	Cellular Compartment (GO domain)
CFinder	Palla et al. overlapping community structure detection
CONOPT	The Constrained Optimization solver
CNM	The Clauset Newman and Moore clustering algorithm
CPLEX	IBM ILOG CPLEX Optimization Studio
CS	Community Strength
CYGD	The Comprehensive Yeast Genome Database

DAG	Directed Acyclic Graph
DICOPT	Discrete and Continuous Optimizer solver
DIP	The Database of Interacting Proteins
EO	The Duch and Arenas Extremal Optimisation algorithm
Exact	Aloise et al. clustering algorithm
GAMS	General Algebraic Modelling System
GN	The Girvan-Newman community structure detection algorithm
GO	The Gene Ontology
IC	Information Content
IP	Integer linear Programming
KEGG	The Kyoto Encyclopedia of Genes and Genomes
Louvain	Blondel et al. clustering method
LP	Linear Programming
MCL	The Markov Clustering Algorithm
MF	Molecular Function (GO domain)
MI	Mutual Information
MICA	Most Informative Common Ancestor
MINLP	Mixed Integer Non Linear Programming
MIPS	The Munich Information Center for Protein Sequences
MIQP	Mixed Integer Quadratic Programming

NLP	Non Linear Programming
NMF	Non-negative Matrix Factorisation
OA	Outer Approximation
OCG	Overlapping Community Generator method from Becker et al.
QP	Quadratic Programming
ORF	Open Reading Frame
PHI-base	The Pathogen-Host Interaction Database
PLEXdb	The Plant Expression Database
PPI	Protein Protein Interaction
QCUT	The spectral clustering algorithm of Ruan and Zhang
RMINLP	Relaxed Mixed Integer Non Linear Programming
SA	The Guimera and Amaral Simulated Annealing clustering algorithm
SBB	The Standard Branch and Bound solver
SGD	The Saccaromyces Genome Database
STRING	The Search Tool for the Retrieval of Interacting Genes/Proteins database
TAP	Tandem Affinity Purification
TC-PINs	Time Course Protein Interaction Networks
UniProt	The Universal Protein Resource database
VV	Verified Virulence
WMI	Within-Module-Interactions

List of Figures

List of Tables